User Guide for servers

# AWS Outposts

# AWS Outposts: User Guide for servers

# Table of Contents

# What is AWS Outposts?

AWS Outposts is a fully managed service that extends AWS infrastructure, services, APIs, and tools to customer premises. By providing local access to AWS managed infrastructure, AWS Outposts enables customers to build and run applications on premises using the same programming interfaces as in AWS Regions, while using local compute and storage resources for lower latency and local data processing needs.

An Outpost is a pool of AWS compute and storage capacity deployed at a customer site. AWS operates, monitors, and manages this capacity as part of an AWS Region. You can create subnets on your Outpost and specify them when you create AWS resources such as EC2 instances and subnets. Instances in Outpost subnets communicate with other instances in the AWS Region using private IP addresses, all within the same VPC.

> **Note**
>
> You cannot connect an Outpost to another Outpost or Local Zone that is within the same VPC.

For more information, see the AWS Outposts product page.

## Key concepts

These are the key concepts for AWS Outposts.

- **Outpost site** – The customer-managed physical buildings where AWS will install your Outpost. A site must meet the facility, networking, and power requirements for your Outpost.
- **Outpost capacity** – Compute and storage resources available on the Outpost. You can view and manage the capacity for your Outpost from the AWS Outposts console.
- **Outpost equipment** – Physical hardware that provides access to the AWS Outposts service. The hardware includes racks, servers, switches, and cabling owned and managed by AWS.
- **Outposts racks** – An Outpost form factor that is an industry-standard 42U rack. Outpost racks include rack-mountable servers, switches, a network patch panel, a power shelf and blank panels.

- **Outposts servers** – An Outpost form factor that is an industry-standard 1U or 2U server, which can be installed in a standard EIA-310D 19 compliant 4 post rack. Outpost servers provide local compute and networking services to sites that have limited space or smaller capacity requirements.
- **Service link** – Network route that enables communication between your Outpost and its associated AWS Region. Each Outpost is an extension of an Availability Zone and its associated Region.
- **Local gateway (LGW)** – A logical interconnect virtual router that enables communication between an Outpost rack and your on-premises network.
- **Local network interface** – A network interface that enables communication from an Outpost server and your on-premises network.

# AWS resources on Outposts

You can create the following resources on your Outpost to support low-latency workloads that must run in close proximity to on-premises data and applications:

**Compute**

| Resource type | Racks | Servers |
|---|---|---|
| Amazon EC2 instances |  Y |  Yes |
| Amazon ECS clusters |  Y |  Yes |
| Amazon EKS nodes |  Y |  No |

### Database and analytics

| Resource type | Racks | Servers |
| --- | --- | --- |
| Amazon ElastiCache nodes (Redis cluster, Memcached cluster) |  Y |  No |
| Amazon EMR clusters |  Y |  No |
| Amazon RDS DB instances |  Y |  No |

### Networking

| Resource type | Racks | Servers |
| --- | --- | --- |
| App Mesh Envoy proxy |  Y |  Yes |
| Application Load Balancers |  Y |  No |
| Amazon VPC subnets |  Y |  Yes |
| Amazon Route 53 |  Y |  No |

**Storage**

| Resource type | Racks | Servers |
| --- | --- | --- |
| Amazon EBS volumes |  Y |  No |
| Amazon S3 buckets |  Y |  No |

**Other AWS services**

| Service | Racks | Servers |
| --- | --- | --- |
| AWS IoT Greengrass |  Y |  Yes |
| Amazon SageMaker Edge Manager |  Y |  Yes |

# Pricing

You can choose from a variety of Outpost configurations, each providing a combination of EC2 instance types and storage options. The price for rack configurations includes installation, removal, and maintenance. For servers, you must install and maintain the equipment.

You purchase a configuration for a 3-year term and can choose from three payment options: All Upfront, Partial Upfront, and No Upfront. If you choose the Partial option or the No Upfront payment option, monthly charges will apply. Any upfront charges apply 24 hours after your Outpost is installed and the compute and storage capacity is available for use. For more information, see:

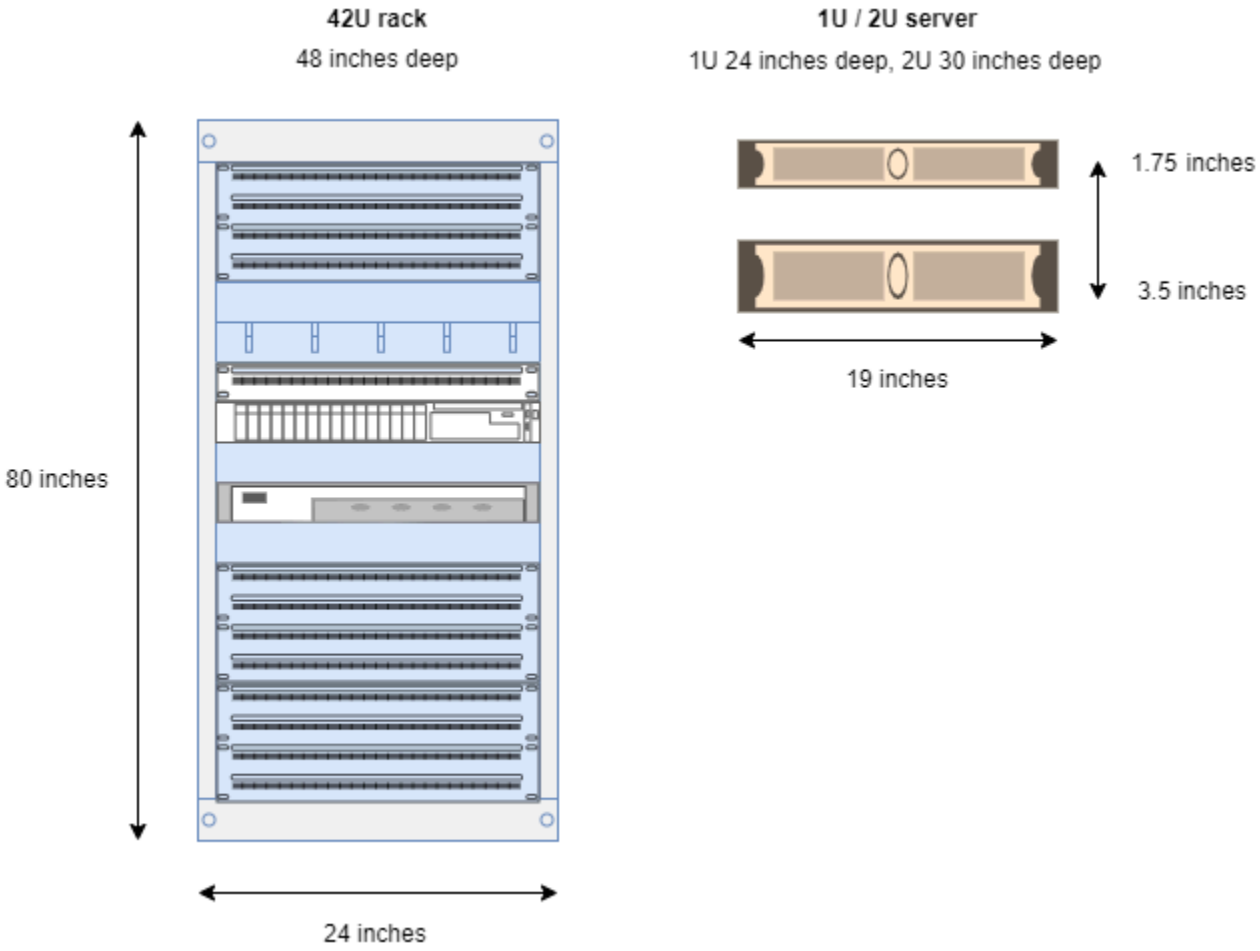
- AWS Outposts rack pricing

- AWS Outposts servers pricing

# How AWS Outposts works

AWS Outposts is designed to operate with a constant and consistent connection between your Outpost and an AWS Region. To achieve this connection to the Region, and to the local workloads in your on-premises environment, you must connect your Outpost to your on-premises network. Your on-premises network must provide wide area network (WAN) access back to the Region and to the internet. It must also provide LAN or WAN access to the local network where your on-premises workloads or applications reside.

The following diagram illustrates both Outpost form factors.




**Contents**

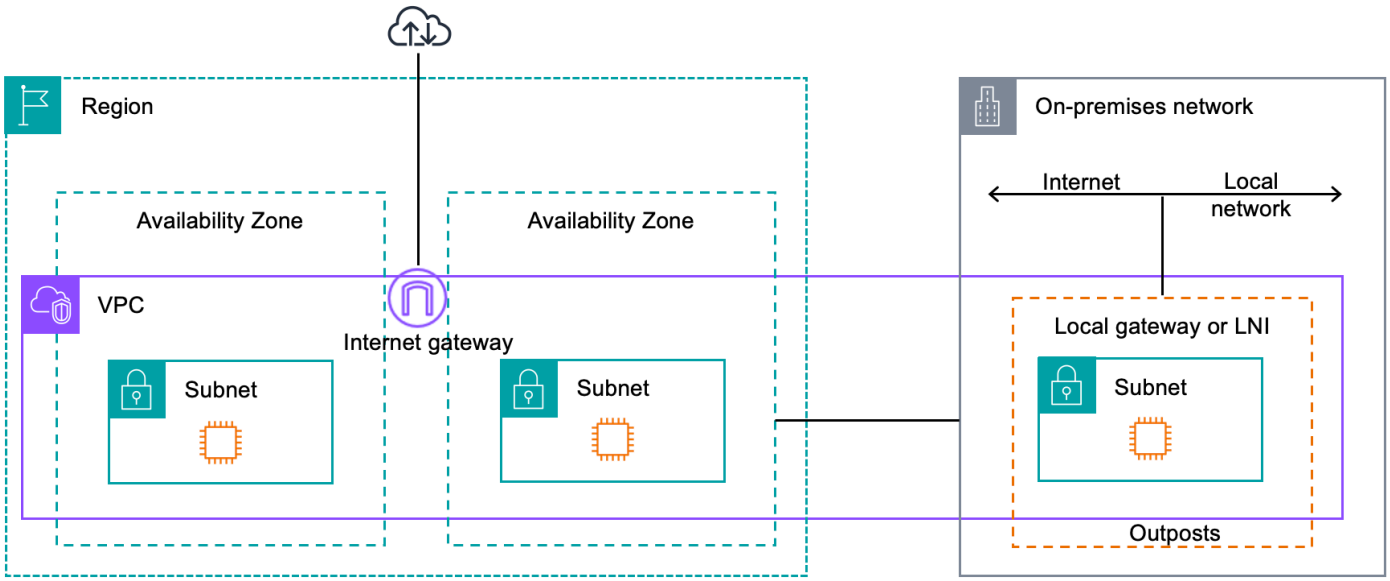
- DNS
- Service link
- Local gateways
- Local network interfaces

# Network components

AWS Outposts extends an Amazon VPC from an AWS Region to an Outpost with the VPC components that are accessible in the Region, including internet gateways, virtual private gateways, Amazon VPC Transit Gateways, and VPC endpoints. An Outpost is homed to an Availability Zone in the Region and is an extension of that Availability Zone that you can use for resiliency.

The following diagram shows the network components for your Outpost.

- An AWS Region and an on-premises network
- A VPC with multiple subnets in the Region
- An Outpost in the on-premises network
- Connectivity between the Outpost and local network provided by either a local gateway (racks) or a local network interface (servers)



# VPCs and subnets

A virtual private cloud (VPC) spans all Availability Zones in its AWS Region. You can extend any VPC in the Region to your Outpost by adding an Outpost subnet. To add an Outpost subnet to a VPC, specify the Amazon Resource Name (ARN) of the Outpost when you create the subnet.

Outposts support multiple subnets. You can specify the EC2 instance subnet when you launch the EC2 instance in your Outpost. You cannot specify the underlying hardware where the instance is deployed, because the Outpost is a pool of AWS compute and storage capacity.

Each Outpost can support multiple VPCs that can have one or more Outpost subnets. For information about VPC quotas, see Amazon VPC Quotas in the *Amazon VPC User Guide*.

You create Outpost subnets from the VPC CIDR range of the VPC where you created the Outpost. You can use the Outpost address ranges for resources, such as EC2 instances that reside in the Outpost subnet.

# Routing

By default, every Outpost subnet inherits the main route table from its VPC. You can create a custom route table and associate it with an Outpost subnet.

The route tables for Outpost subnets work as they do for Availability Zone subnets. You can specify IP addresses, internet gateways, local gateways, virtual private gateways, and peering connections as destinations. For example, each Outpost subnet, either through the inherited main route table, or a custom table, inherits the VPC local route. This means that all traffic in the VPC, including the Outpost subnet with a destination in the VPC CIDR remains routed in the VPC.

Outpost subnet route tables can include the following destinations:

- **VPC CIDR range** – AWS defines this at installation. This is the local route and applies to all VPC routing, including traffic between Outpost instances in the same VPC.
- **AWS Region destinations** – This includes prefix lists for Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB gateway endpoint, AWS Transit Gateways, virtual private gateways, internet gateways, and VPC peering.

  If you have a peering connection with multiple VPCs on the same Outpost, the traffic between the VPCs remains in the Outpost and does not use the service link back to the Region.

# DNS

For network interfaces connected to a VPC, EC2 instances in Outposts subnets can use the Amazon Route 53 DNS Service to resolve domain names to IP addresses. Route 53 supports DNS features, such as domain registration, DNS routing, and health checks for instances running in your Outpost. Both public and private hosted Availability Zones are supported for routing traffic to specific domains. Route 53 resolvers are hosted in the AWS Region. Therefore, service link connectivity from the Outpost back to the AWS Region must be up and running for these DNS features to work.

You might encounter longer DNS resolution times with Route 53, depending on the path latency between your Outpost and the AWS Region. In such cases, you can use the DNS servers installed locally in your on-premises environment. To use your own DNS servers, you must create DHCP option sets for your on-premises DNS servers and associate them with the VPC. You must also ensure that there is IP connectivity to these DNS servers. You might also need to add routes to the local gateway routing table for reachability but this is only an option for Outpost racks with local gateway. Because DHCP option sets have a VPC scope, instances in both the Outpost subnets and the Availability Zone subnets for the VPC will try to use the specified DNS servers for DNS name resolution.

Query logging is not supported for DNS queries originating from an Outpost.

# Service link

The service link is a connection from your Outpost back to your chosen AWS Region or Outposts home Region. The service link is an encrypted set of VPN connections that are used whenever the Outpost communicates with your chosen home Region. You use a virtual LAN (VLAN) to segment traffic on the service link. The service link VLAN enables communication between the Outpost and the AWS Region for both management of the Outpost and intra-VPC traffic between the AWS Region and Outpost.

Your service link is created when your Outpost is provisioned. If you have a server form factor, you create the connection. If you have a rack, AWS creates the service link. For more information, see Outpost connectivity to AWS Regions.

# Local gateways

Outpost racks include a local gateway to provide connectivity to your on-premises network. If you have an Outpost rack, you can include a local gateway as target where the destination is your on-

premises network. Local gateways are only available for Outpost racks and can only be used in VPC and subnet route tables that are associated with an Outpost rack. For more information, see Local gateway in the *AWS Outposts User Guide for Outposts rack*.

## Local network interfaces

Outpost servers include a local network interface to provide connectivity to your on-premises network. A local network interface is available only for Outposts servers running on an Outpost subnet. You cannot use a local network interface from an EC2 instance on an Outpost rack or in the AWS Region. The local network interface is meant only for on-premises locations. For more information, see Local network interfaces.

# Site requirements for Outposts servers

An Outpost site is the physical location where your Outpost operates. Sites are only available in select countries and territories. For more information, see AWS Outposts servers FAQs. Refer to the question: **In which countries and territories are Outposts servers available?**

This page covers the requirements for Outposts servers. For the requirements for Outposts rack, see Site requirements for Outposts rack in the *AWS Outposts User Guide for Outposts rack*.

## Facility

These are the facility requirements for servers.

> **Note**
>
> Specifications are for servers under normal operating conditions. For example, acoustics may sound louder during initial installation and then operate at the rated sound power after installation is complete.

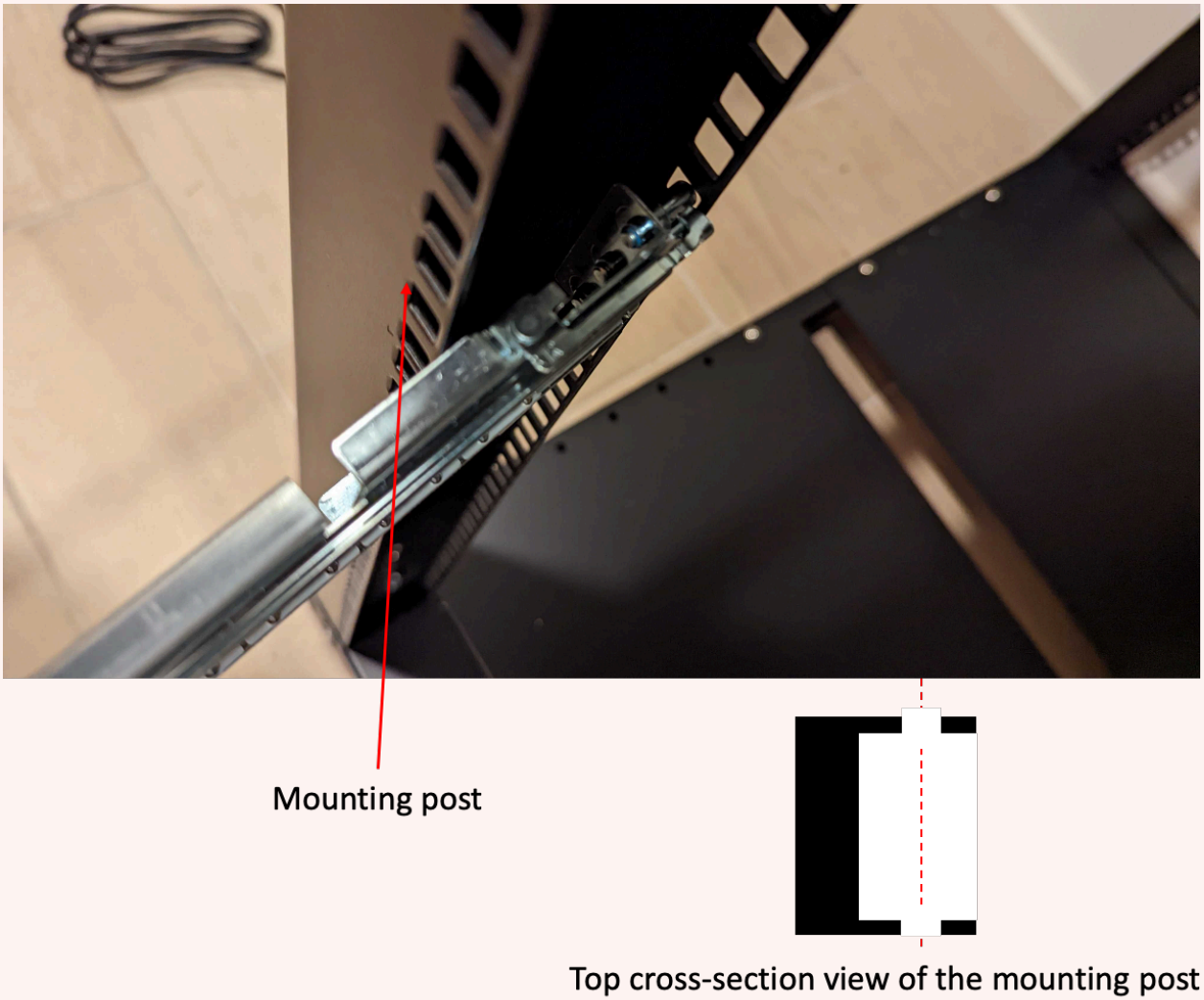
- **Temperature** – The ambient temperature must be between 41–95° F (5–35° C).

  The server will shut down when the temperature is outside this range and will restart when the temperature is back within the range.
- **Humidity** – The relative humidity must be between 8–80 percent with no condensation.
- **Air quality** – The air must be filtered using a MERV8 (or higher) filter.
- **Airflow** – The position of the server must ensure a minimum clearance of 6 inches (15 cm) between the server and walls in front of and behind the server to allow for sufficient airflow clearance.
- **Weight** – The 1U server weighs 26 pounds and the 2U server weighs 36 pounds. Confirm that the location where you intend to put the server can support the weight of the server.

  To see the weight requirements for different Outposts resources, choose **Browse catalog** in the AWS Outposts console at https://console.aws.amazon.com/outposts/.
- **Rail-kit compatibility** – The rail kit that is included in your shipping package is compatible with a standard L-shaped mounting bracket of an EIA-310-D compliant 19 inch rack.

**Important**

The rail kit is **not** compatible with a U-shaped mounting bracket as shown in the following image.




- **Rack Placement** – We recommend the use of standard 19-inch EIA-310D racks, with a depth of at least 36 inches (914 mm).
  - Outposts 2U servers require space with the following dimensions: 3.5 inches height (88.9mm), 17.5 inches width (447 mm), 30 inches depth (762 mm)
  - Outposts 1U servers require space with the following dimensions: 1.75 inches height (44.45 mm), 17.5 inches width (447 mm), 24 inches depth (610 mm)

> **Note**
>
> - Mounting AWS Outposts servers vertically is not supported.
> - Outposts 1U servers are the same width as Outposts 2U servers, but half the height and less depth

AWS provides a rail kit for rack-mounting the server. For more information, see Step 3: Rack mount.

If you do not place the server in a rack, you must still meet the other requirements listed in this section.

- **Serviceability** – Outposts servers are front-aisle serviceable.
- **Acoustics** – rated to be less than 78 dBA sound power at temperatures of 80 ° F (27 ° C) and meets GR-63 CORE NEBS compliance.
- **Seismic bracing** – To the extent required by regulation or code, you will install and maintain appropriate seismic anchorage and bracing for the server while it is in your facility.
- **Elevation** – The elevation of the room where the rack is installed must be below 10,005 feet (3,050 meters).
- **Cleaning** – Wipe surfaces with damp wipes that contain approved antistatic cleaning chemicals.

## Networking

Each Outposts server includes non-redundant physical uplink ports. Ports have their own speed and connector requirements as detailed below.

| Port label | Speed | Connector on the upstream networkin g device | Traffic |
|---|---|---|---|
| Port 3 | 10Gbe | SFP+ | Both service and LNI link traffic – QSFP+ breakout cable (10 feet/3 m) segments traffic. For |

| Port label | Speed | Connector on the upstream networking device | Traffic |
|---|---|---|---|
| | | | more information, see Configure QSFP network. |

## Service link firewall

UDP and TCP 443 must be statefully listed in the firewall.

| Protocol | Source Port | Source Address | Destination Port | Destination Address |
|---|---|---|---|---|
| UDP | 1024-65535 | Service Link IP | 53 | DHCP provided DNS server |
| UDP | 443, 1024-65535 | Service Link IP | 443 | Outposts Service Link endpoints |
| TCP | 1024-65535 | Service Link IP | 443 | Outposts Registration endpoints |

You can use an AWS Direct Connect connection or a public internet connection to connect the Outpost back to the AWS Region. For Outposts service link connectivity, you can use NAT or PAT at your firewall or edge router. Service link establishment is always initiated from the Outpost.

## Service link maximum transmission unit (MTU)

The network must support 1500-bytes MTU between the Outpost and the service link endpoints in the parent AWS Region. For more information about the service link, see AWS Outposts connectivity to AWS Regions.

## Service link bandwidth recommendations

For an optimal experience and resiliency, AWS recommends that you use redundant connectivity of at least 500 Mbps for the service link connection to the AWS Region. The maximum utilization for

each Outpost server is 500 Mbps. To increase the connection speed, use multiple Outpost servers. For example, if you have three AWS Outposts servers, the maximum connection speed increases to 1.5 Gbps (1,500 Mbps). For more information, see Service link traffic for servers.

Your AWS Outposts service link bandwidth requirements vary depending on workload characteristics, such as AMI size, application elasticity, burst speed needs, and Amazon VPC traffic to the Region. Note that AWS Outposts servers do not cache AMIs. AMIs are downloaded from the Region with every instance launch.

To receive a custom recommendation about the service link bandwidth required for your needs, contact your AWS sales representative or APN partner.

### Service link requires DHCP response

The service link requires an IPv4 DHCP response to configure network settings.

### Service link maximum latency

Service links can support up to a maximum network latency of 250 ms from the server and its Availability Zone.

## Power

These are the power requirements for Outposts servers.

**Requirements**

- Power support
- Power draw
- Power cable
- Power redundancy

### Power support

Servers are rated up to 1600W 90-264 VaC 47/63 Hz AC power.

### Power draw

To see the power draw requirements for different Outposts resources, choose **Browse catalog** in the AWS Outposts console at https://console.aws.amazon.com/outposts/.

## Power cable

The server ships with an IEC C14-C13 power cable.

**Power cabling from server to rack**

Use the provided IEC C14-C13 power cable to connect the server to the rack.

**Power cabling from server to wall outlet**

To connect the server to a standard wall outlet, you must use either an adapter for the C14 inlet or a country-specific power cord.

Ensure that you have the correct adapter or power cord for your region to save time during server installation.

- In the United States, you need an IEC C13 to NEMA 5-15P power cord.
- In parts of Europe, you might need an IEC C13 to CEE 7/7 power cord.
- In India, you need an IEC C13 to IS1293 power cord.

## Power redundancy

Servers include multiple power connections and ship with cables to enable power redundant operation. We recommend power redundancy, but redundancy is not required.

Servers do not include an Uninterruptible Power Supply (UPS).

# Order fulfillment

To fulfill the order, AWS will ship the Outposts server equipment, including rail mounts and required power and network cables, to the address that you provided. The box that the server is shipped in has the following dimensions:

- Box with a 2U server:
  - Length: 44 inches / 111.8 cm
  - Height: 26.5 inches / 67.3 cm
  - Width: 17 inches / 43.2 cm
- Box with a 1U server:

- Length: 34.5 inches / 87.6 cm
- Height: 24 inches / 61 cm
- Width: 9 inches / 22.9 cm

Your team or a third-party provider must install the equipment. For more information, see Outpost server installation.

The installation is complete when you confirm that the Amazon EC2 capacity for your Outposts server is available from your AWS account.

# Get started with AWS Outposts

Order an Outpost to get started. After installation of your Outpost equipment, launch Amazon EC2 instances and access your on-premises network.

**Tasks**

- Create an Outpost and order Outpost capacity
- Outpost server installation
- Launch an instance on your Outpost server

# Create an Outpost and order Outpost capacity

To begin using AWS Outposts, log in with the AWS account that will own the Outpost. Create a site and an Outpost. Then, place an order for the Outposts servers that you require.

**Prerequisites**

- Review the available configurations for your Outposts servers.
- An Outpost site is the physical location for your Outpost equipment. Before ordering capacity, verify that your site meets the requirements. For more information, see Site requirements for Outposts servers.
- You must have an AWS Enterprise Support plan.
- Determine which AWS account will own the Outpost. Use this account to create the Outposts site, create the Outpost, and place the order. Monitor the email associated with this account for information from AWS.

**Tasks**

- Step 1: Create a site
- Step 2: Create an Outpost
- Step 3: Place the order
- Step 4: Modify instance capacity
- Next steps

## Step 1: Create a site

Create a site to specify the operating address. The operating address is the location where you will install and run your Outposts servers. After you create the site, AWS Outposts assigns an ID to your site. You must specify this site when you create an Outpost.

**Prerequisites**

- Determine the operating address.

**To create a site**

1. Sign in to AWS using the AWS account that will own the Outpost.
2. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
3. To select the parent AWS Region, use the Region selector in the upper-right corner of the page.
4. In the navigation pane, choose **Sites**.
5. Choose **Create site**.
6. For **Supported hardware type**, choose **Servers only**.
7. Enter the name, description, and operating address for your site.
8. (Optional) For **Site notes**, enter any other information that might be useful for AWS to know about the site.
9. Choose **Create site**.

## Step 2: Create an Outpost

Create an Outpost for each server. An Outpost can only be associated with a single server. You'll specify this Outpost when you place the order.

**Prerequisites**

- Determine the AWS Availability Zone to associate with your site.

**To create an Outpost**

1. In the navigation pane, choose **Outposts**.

2. Choose **Create Outpost**.
3. Choose **Servers**.
4. Enter the name and a description for your Outpost.
5. Choose an Availability Zone for your Outpost.
6. For **Site ID**, choose your site.
7. Choose **Create Outpost**.

## Step 3: Place the order

Place an order for the Outposts racks that you need. After you submit the order, an AWS Outposts representative will contact you.

> **Important**
>
> You can't edit an order after you submit it so review all details carefully before submission. If you need to change an order, contact your AWS Account Manager.

### Prerequisites

- Determine how you will pay for the order. You can pay all upfront, partially upfront, or nothing upfront. If you choose the partial-upfront or no-upfront payment option, you'll pay monthly charges over the three-year term.

  The pricing includes delivery, infrastructure service maintenance, and software patches and upgrades.
- Determine whether the shipping address is different from the operating address that you specified for the site.

### To place an order

1. In the navigation pane, choose **Orders**.
2. Choose **Place order**.
3. For **Supported hardware type**, choose **Servers**.
4. To add capacity, choose a configuration.

5. Choose **Next**.
6. Choose **Use an existing Outpost** and select your Outpost.
7. Choose **Next**.
8. Select a contract term and payment option.
9. Specify the shipping address. You can specify a new address or select the site's operating address. If you select the operating address, be aware that any future change to the site's operating address will not propagate to existing orders. If you need to change the shipping address on an existing order, contact your AWS Account Manager.
10. Choose **Next**.
11. On the **Review and order** page, verify that your information is correct and edit as needed. You will not be able to edit the order after you submit it.
12. Choose **Place order**.

## Step 4: Modify instance capacity

The capacity of each new Outpost order is configured with a default capacity configuration. You can convert the default configuration to create various instances to meet your business needs. To do so, you create a capacity task, specify the instance sizes and quantity, and run the capacity task to implement the changes.

> **Note**
>
> - You can change the quantity of instance sizes after you place the order for your Outposts.
> - Instances sizes and quantities are defined at the Outpost level.
> - Instances are placed automatically based on best practices.

### To modify instance capacity

1. From the AWS Outposts console'sAWS Outposts left navigation pane, choose **Capacity tasks**.
2. On the **Capacity tasks** page, choose **Create capacity task**.
3. On the **Getting started** page, choose the order.
4. To modify capacity, you can use the steps in the console or upload a JSON file.

Console steps

1. Choose **Modify a new Outpost capacity configuration**.
2. Choose **Next**.
3. On the **Configure instance capacity** page, each instance type shows one instance size with the maximum quantity preselected. To add more instance sizes, choose **Add instance size**.
4. Specify the instance quantity and note the capacity that is displayed for that instance size.
5. View the message at the end of each instance-type section that informs you if you are over or under capacity. Make adjustments at the instance size or quantity level to optimize your total available capacity.
6. You can also request AWS Outposts to optimize the instance quantity for a specific instance size. To do so:

   a. Choose the instance size.

   b. Choose **Auto-balance** at the end of the related instance-type section.
7. For each instance type, ensure that the instance quantity is specified for at least one instance size.
8. Choose **Next**.
9. On the **Review and create** page, verify the updates that you are requesting.
10. Choose **Create**. AWS Outposts creates a capacity task.
11. On the capacity task page, monitor the status of the task.

> **Note**
>
> AWS Outposts might request you to stop one or more running instances to enable running the capacity task. After you stop these instances, AWS Outposts will run the task.

Upload JSON file

1. Choose **Upload a capacity configuration**.
2. Choose **Next**.
3. On the **Upload capacity configuration plan** page, upload the JSON file that specifies the instance type, size, and quantity.

**Example**

Example JSON file:

```
{
    "RequestedInstancePools": [
        {
            "InstanceType": "c5.24xlarge",
            "Count": 1
        },
        {
            "InstanceType": "m5.24xlarge",
            "Count": 2
        }
    ]
}
```

4. Review the contents of the JSON file in the **Capacity configuration plan** section.
5. Choose **Next**.
6. On the **Review and create** page, verify the updates that you are requesting.
7. Choose **Create**. AWS Outposts creates a capacity task.
8. On the capacity task page, monitor the status of the task.

> **Note**
>
> AWS Outposts might request you to stop one or more running instances to enable running the capacity task. After you stop these instances, AWS Outposts will run the task.

## Next steps

You can view the status of your order using the AWS Outposts console. The initial status of your order is **Order received**. An AWS representative will contact you within three business days. You will receive an email confirmation when the status of your order changes to **Order processing**. An AWS representative may contact you to get any additional information that AWS requires.

If you have any questions about your order, contact AWS Support.

To fulfill the order, AWS will schedule a delivery date.

You are responsible for all installation tasks, including physical installation and network configuration. You can contract a third-party to perform these tasks for you. Whether you do the installation or contract to a third-party, installation requires IAM credentials in the AWS account that contains the Outpost to verify the identity of the new device. You are responsible for providing and managing this access. For more information, see the section called “Outpost server installation”.

The installation is complete when Amazon EC2 capacity for your Outpost is available from your AWS account. After the capacity is available, you can launch Amazon EC2 instances on your Outpost server. For more information, see the section called “Launch an instance”.

## Outpost server installation

When you order an Outpost server, you are responsible for installation, whether you do it yourself or contract a third party. The party installing requires specific permissions to verify the identity of the new device. For more information, see Grant permissions.

**Prerequisite**

You must have an Outpost server form factor at your site. For more information, see Create an Outpost and order Outpost capacity.

> **Note**
>
> We recommend that you view the Installing AWS Outposts Servers training video before and during the installation process. To access the training, you must sign in or create an account on AWS Skill Builder.

**Tasks**

- Step 1: Grant permissions
- Step 2: Inspect
- Step 3: Rack mount
- Step 4: Power up
- Step 5: Connect network
- Step 6: Authorize server

- Outpost Configuration Tool command reference

## Step 1: Grant permissions

To verify the identity of the new device, you must have IAM credentials in the AWS account that contains the Outpost. The AWSOutpostsAuthorizeServerPolicy policy grants the permissions required to install an Outpost server. For more information, see the section called "Identity and access management".

### Considerations

- If you are using a third party that does not have access to your AWS account, you must provide temporary access.
- AWS Outposts supports using temporary credentials. You can configure temporary credentials that last up to 36 hours. Ensure that you give the installer enough time to perform all the steps for server installation. For more information, see the section called "Temporary credentials".

## Step 2: Inspect

To complete an inspection of the Outposts equipment, you should check the shipping package for damage, unpack the shipping package, and locate the Nitro Security Key (NSK). Consider the following information about inspecting the server:

- The shipping package has shock sensors located on the two largest sides of the box.
- The inside flap of the shipping package contains instructions about how to unpack the server and locate the NSK.
- The NSK is an encryption module. To complete inspection, you *locate* the NSK. You attach the NSK to the server in a later step.

### Check the shipping package

### To inspect the shipping package

- Before you open the shipping package, observe both shock sensors and note if they have been activated. If the shock sensors have been activated it is possible that the unit has been damaged. Proceed with the installation taking time to note any further damage to the server

or accessories. If any part of the system is obviously damaged or the installation fails to proceed as expected contact AWS Support for guidance on replacing your Outposts server.


If the bar in the middle of the sensor is red, the sensor has been activated.

### Unpack the shipping package

### To unpack the shipping package

- Open the package and ensure it contains the following items:
  - Server
  - Nitro Security Key (encryption module) – packaging marked with "NSK" in red. See the following procedure for locating the NSK from the shipping package for more information.
  - Rack installation kit (2 inner rails, 2 outer rails, and screws)
  - Installation pamphlet
  - Accessory kit
    - Pair of C13/14 power cables - 10 feet (3m)

- QSFP breakout cable -10 feet (3m)
- USB cable, micro-USB to USB-C - 10 feet (3m)
- Brush guard

**Find the NSK**

The NSK is inside the box labelled **A** that includes the accessories for the server.

> **Important**
>
> Do not use the NSK to destroy data on the server during installation.

The NSK is required to activate the server. The NSK is also used to destroy data on the server when you send the server back. In this installation step, **ignore** the instructions on the body of the NSK as those instructions are to destroy data.

## Step 3: Rack mount

To complete this step, you must attach inner rails to the server, outer rails to the rack, then mount the server on the rack. You need a Phillips-head screwdriver to complete these steps.

**Rack mount alternatives**

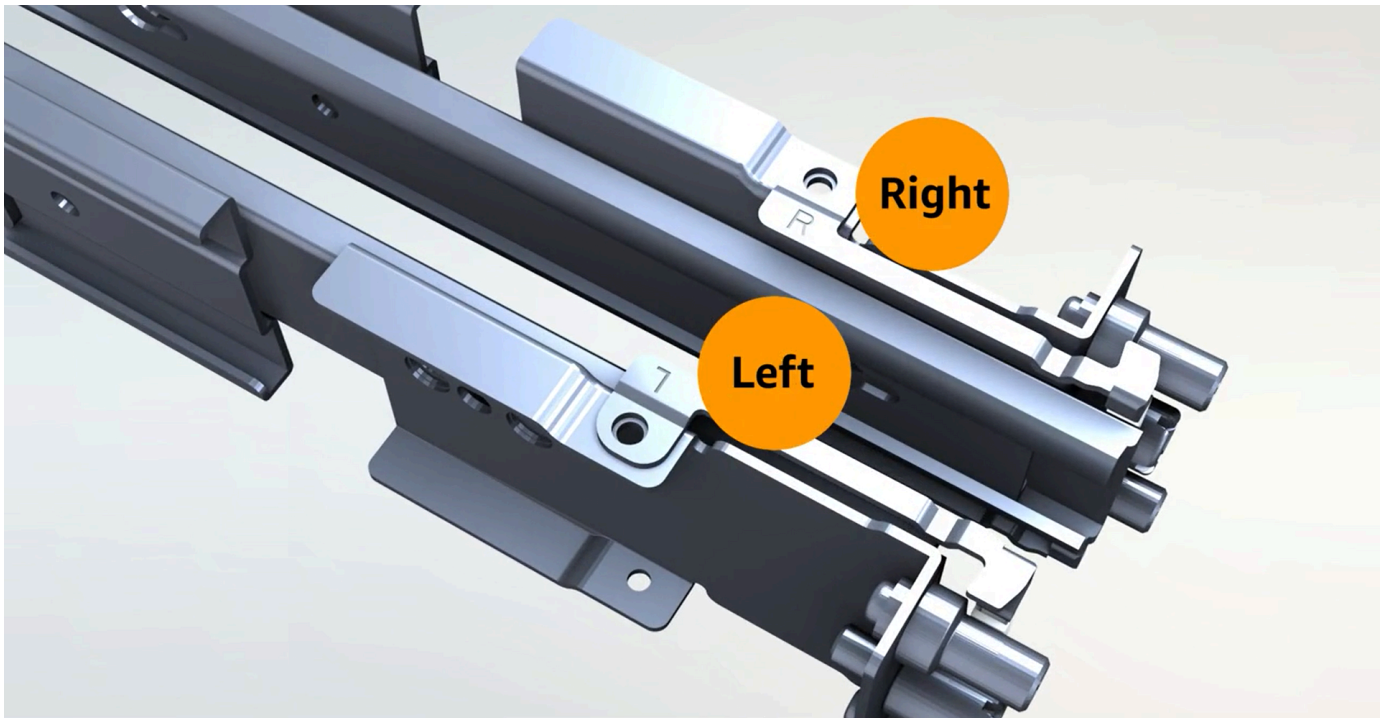
You are not required to mount the server in a rack. If you're not mounting the server in a rack, consider the following information:

- Ensure a minimum clearance of 6 inches (15 cm) between the server and walls in front of and behind the server to allow the hot air to circulate.
- Place the server on a stable surface free from mechanical hazards such as moisture or falling objects.
- To use the networking cables included with the server, you must place the server within 10 feet (3 m) of your upstream networking device.
- Follow local guidance for seismic bracing and bonding.

### Identify sides and ends

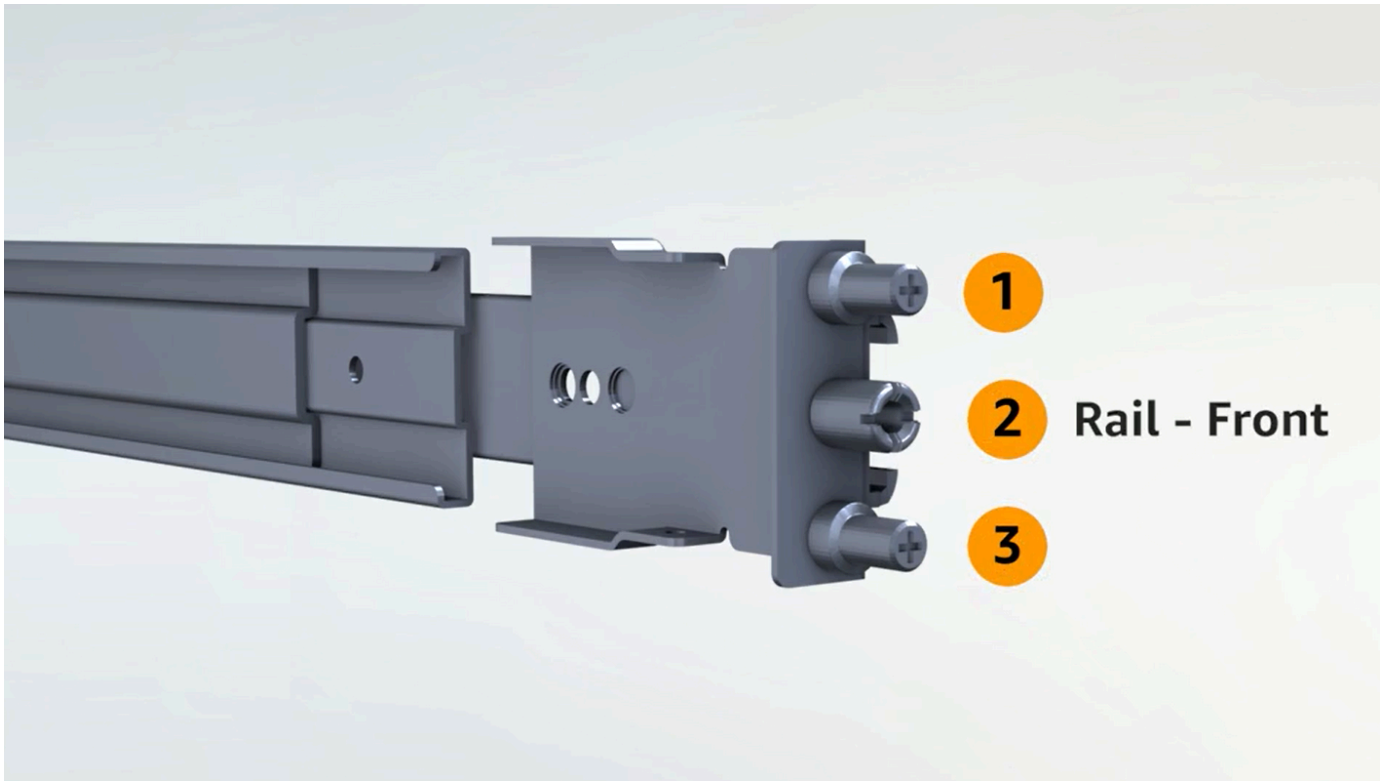
### To identify left from right, front from back

1. Locate and open the box of rack rails that came with the server.
2. Look at the markings on the rails to determine which is left and right. These markings determine to which side of the server each rail gets attached.

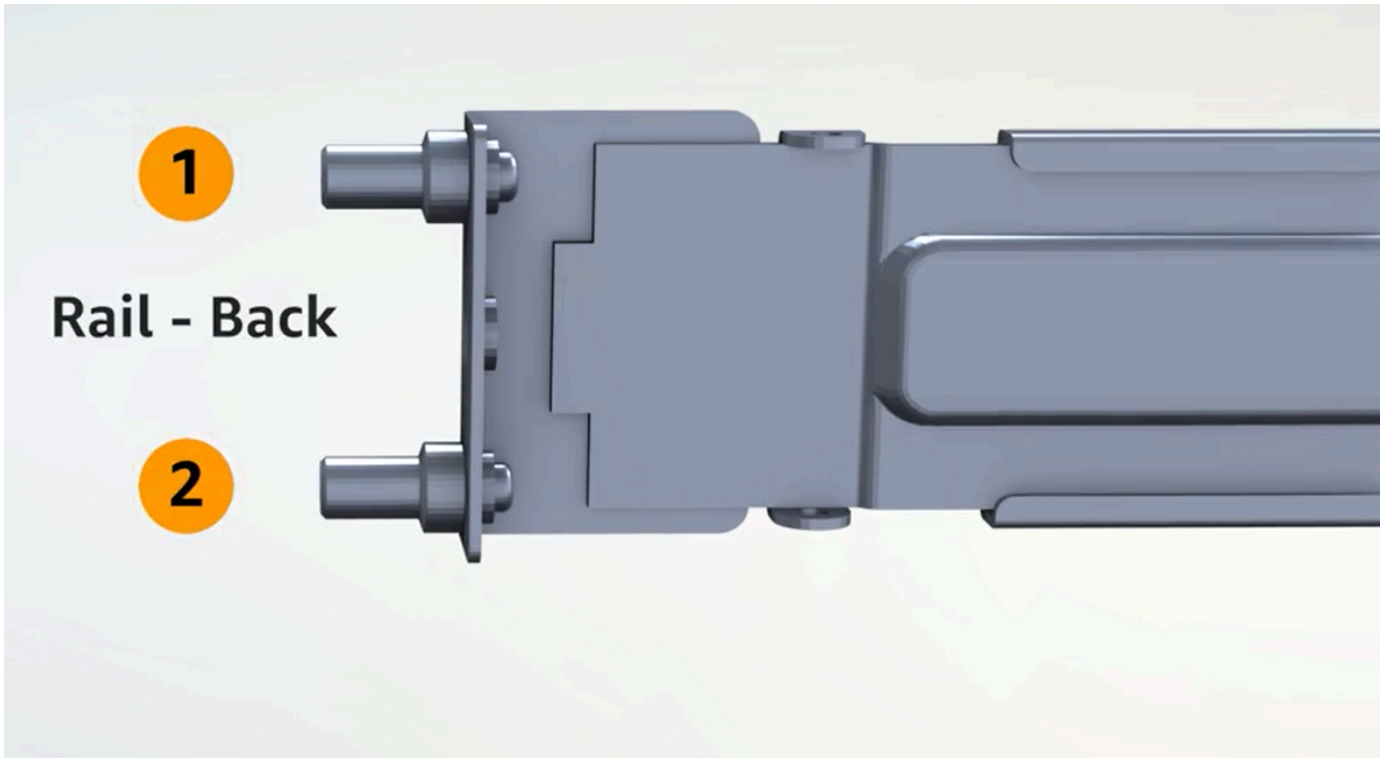



3. Look at the posts on each end of the rails to determine which is front, and which is back.

   The front end has three posts.




The back end has two posts.



### Attach inner rails

#### To attach inner rails to the server

1. Detach the inner rail from the outer rail for both rails. You should have four rails.
2. Attach the right inner rail to the right side of the server and secure the rail with a screw. Make sure you orient the rail correctly with the server. Point the front part of the rail toward the front of the server.
3. Attach the left inner rail to the left side of the server and secure the rail with a screw.

### Attach outer rails

#### To attach outer rails to the rack

1. Face the rack and use the rail marked R on the right side of the rack. Attach the back of the rail to the rack first, then extend the rail to connect it to the front of the rack.

   > **Tip**
   >
   > Pay attention to the orientation of the rails. Use included pin adapters if necessary.

2. Repeat with the left rail on the left side.

### Mount the server

#### To mount the server in the rack

- Slide the server into the outer rails you installed on the rack in the previous step and secure the server at the front with two provided screws.

  > **Tip**
  >
  > Use two people to slide the server into the rack.

# Step 4: Power up

To complete power up, you attach the NSK, connect the server to a power source, and verify that the server has powered on. Consider the following information about powering the server:

- The server functions with one power source, but AWS recommends you use two power sources for redundancy.
- Connect the power cables before you connect the network cables.
- Use the pair of C13 outlet/C14 inlet power cables to connect the server to a power supply on the rack. If you're not using the C14 inlet power cable to connect the server to a power supply on the rack, you must provide adapters for the C14 inlets that connect to a power source.

### Attach NSK

You must attach the NSK to the server so it can decrypt data on the server during operation.

**Important**

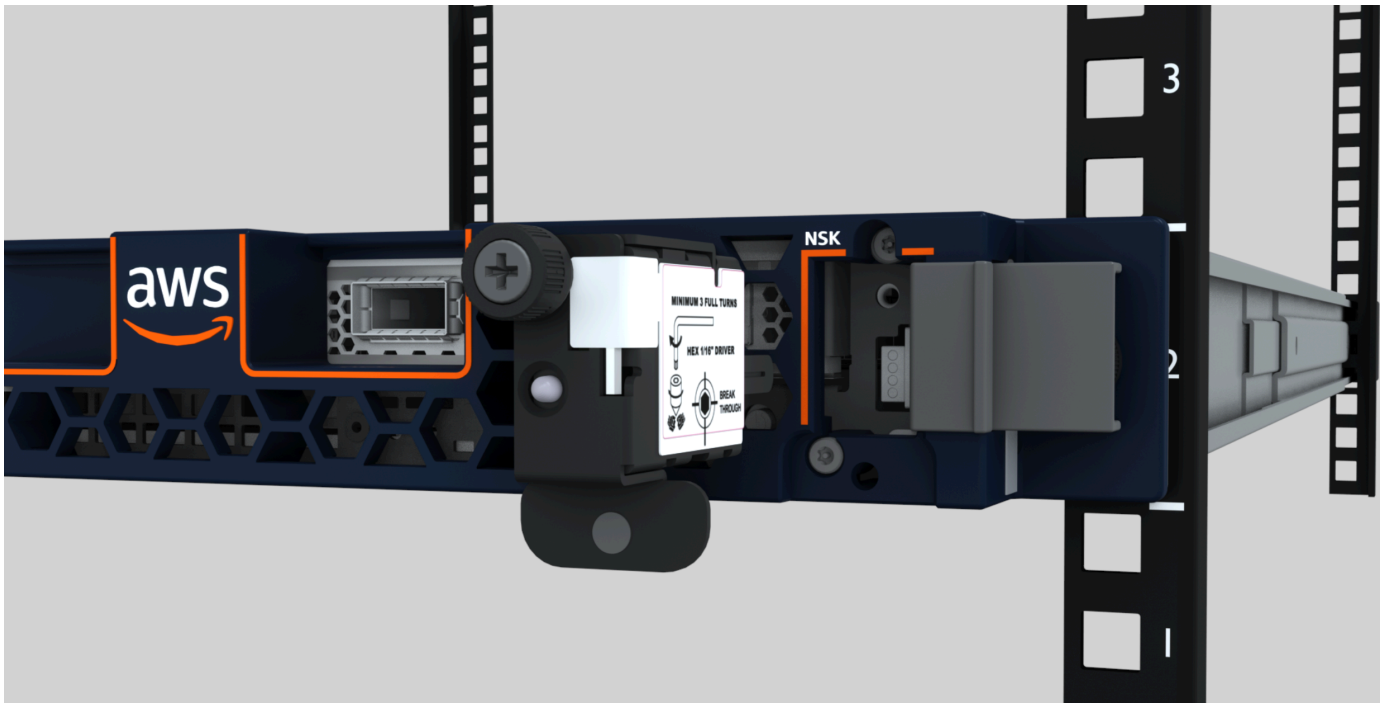
- The side of the NSK has instructions on how to destroy the NSK. Do not follow those instructions now. Follow those instructions only when returning the server to AWS, to cryptographically shred the data on the server.
- If you are installing multiple servers at the same time, ensure that you do not mix up the NSKs. You must attach the NSK to the server that it shipped with. If you use a different NSK, the server will not boot up.

### To attach the NSK

1. On the front right side of the server, open the NSK compartment.

   The following image shows the NSK attached to a 2U server.


The following image shows the NSK attached to a 1U server.

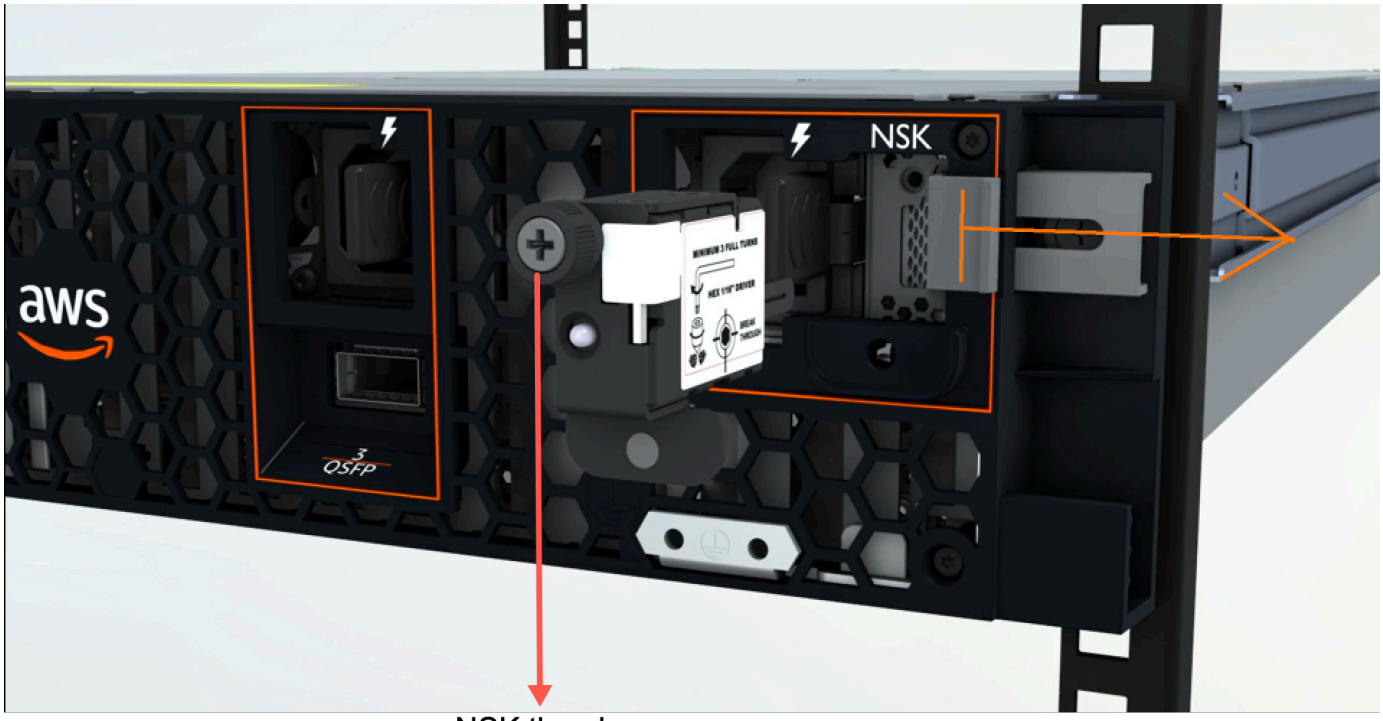



2. Ensure that the serial number (SN) on the NSK matches the SN on the bezel pull-out tab of the NSK compartment on the server.

   The following image shows the SN number on the NSK and bezel pull-out tab:


3. Fit the NSK into the slot.
4. Hand tighten using the thumbscrew or tighten with a screwdriver (0.7 Nm / 0.52 lb-ft) until snug. Do not use power tool as it might over-torque and damage the NSK.

   The following image shows the location of the thumbscrew.




NSK thumbscrew

The following image shows the type of screwdriver you can use to attach the NSK to the server.

## Power up

### To connect the server to power

1. Locate the pair of C13/C14 power cables that came with the server.
2. Connect the C14 end of both cables to your power source.
3. Connect the C13 end of both cables to the ports on the front of the server.

## Verify server power

### To verify that the server has power

1. Verify that you can hear the server running.

   > **Tip**
   >
   > The noise level goes down after the server provisions itself.

2. Verify that the LED power lights above the power ports are lit.

   The following image shows the LED power lights on a 2U server

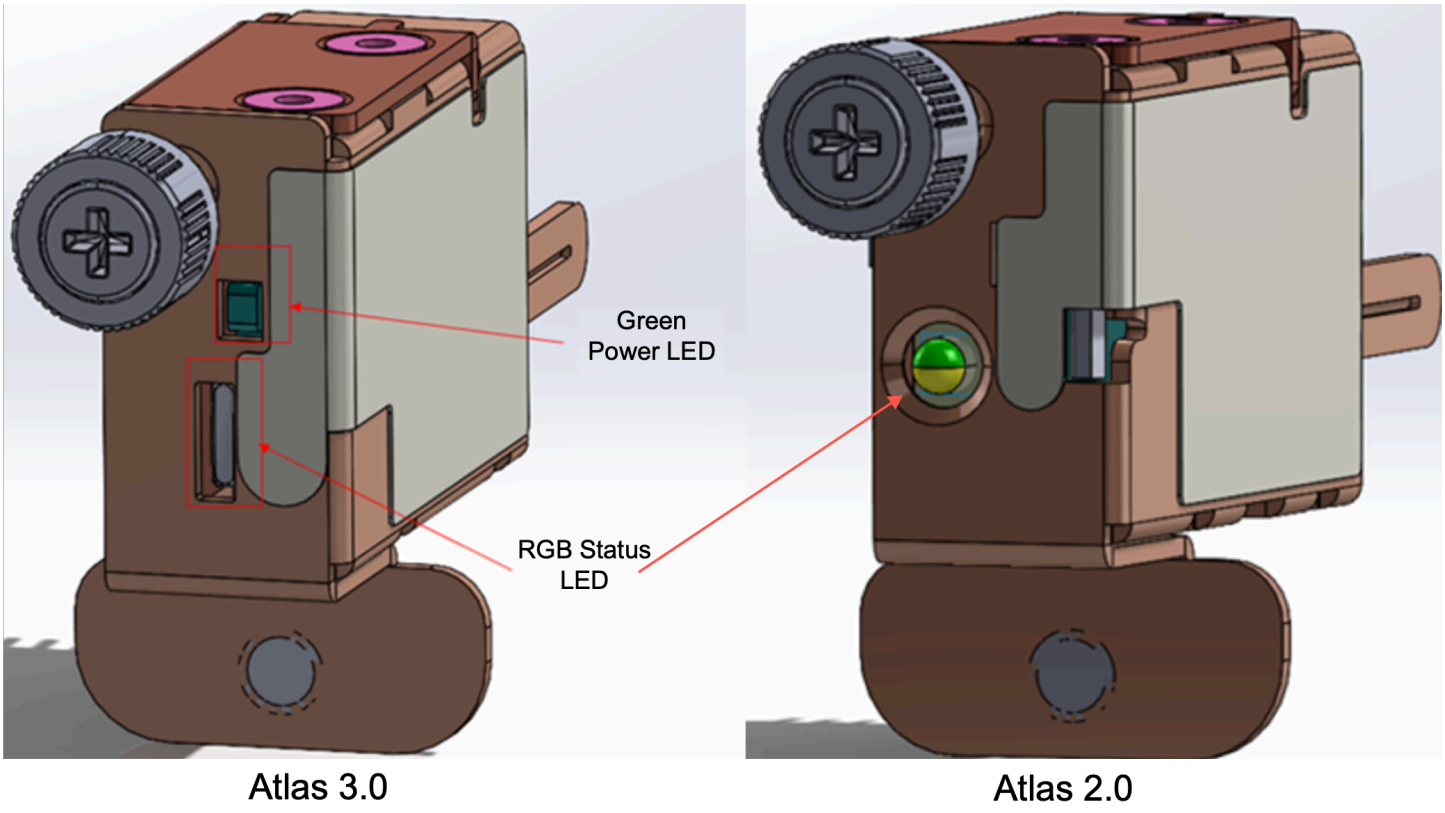
   
   

   The following image shows the LED power lights on a 1U server

### Check the Power LED on the Atlas 3.0. NSK

AWS Outposts supports two versions of NSK: Atlas 2.0 and Atlas 3.0. Both NSK versions have a RGB **Status** LED. In addition, the Atlas 3.0 has a green **Power** LED. This step is only for the Atlas 3.0 NSK.

The following image shows the location of the LEDs on the Atlas 2.0 and Atlas 3.0 NSKs:



If you have the Atlas 2.0 NSK, skip to the next step, Step 5: Connect network because this version of the NSK only has the RGB Status LED which you must check after the Outpost server is provisioned and activated.

If you have the Atlas 3.0 NSK, check the green Power LED:

- If the green light is on, the NSK is correctly connected to the host and has power. You can proceed to the next step.
- If the green light is off, the NSK is not correctly connected to the host or/and has no power. Contact AWS Support.

## Step 5: Connect network

To complete the network setup, you connect the server to your upstream networking device with network cable.

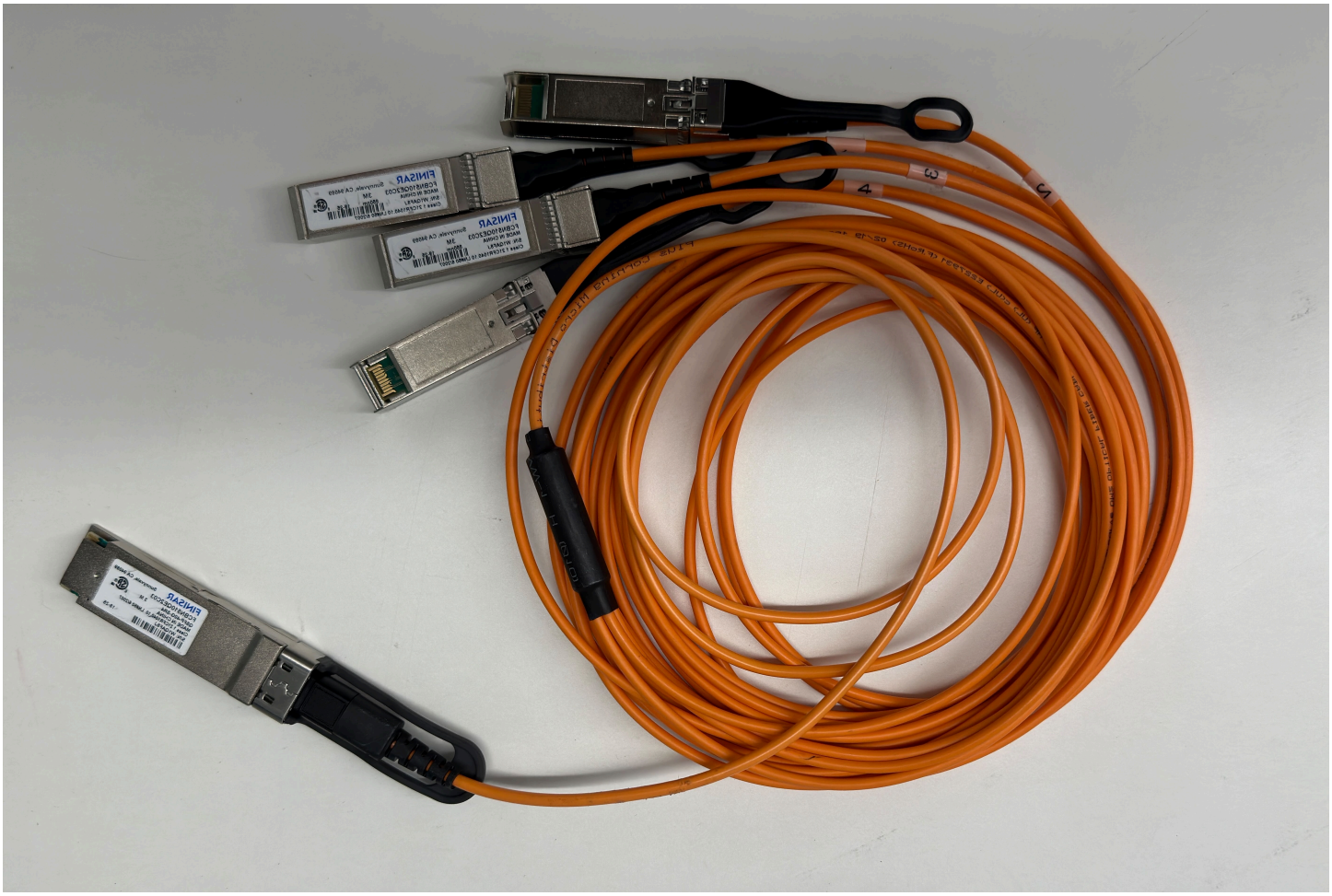
Consider the following information about connecting to the network:

- The server requires connections for two types of traffic: service link traffic and local network interface (LNI) link traffic. The instructions in the following section describe which ports to use on the server to segment traffic. Consult with your IT group to determine which port on your upstream networking device should carry each type of traffic.
- Ensure the server has connected to your upstream networking device and has been assigned an IP address. For more information, see Server IP address assignment.
- The optical connection on an AWS Outposts server only supports 10 Gbits and does not support auto-negotiation of port speed. If the host port tries to negotiate port speed, for example, between 10 through 25 Gbits, you can run into problems. In such cases, we recommend you do the following:
  - Set the port speed on the switch port to 10 Gbits.
  - Work with your switch vendor to support a static configuration.

### Configure QSFP network

With the QSFP breakout cable, you use breakouts to segment traffic.

The following image shows the QSFP breakout cable:

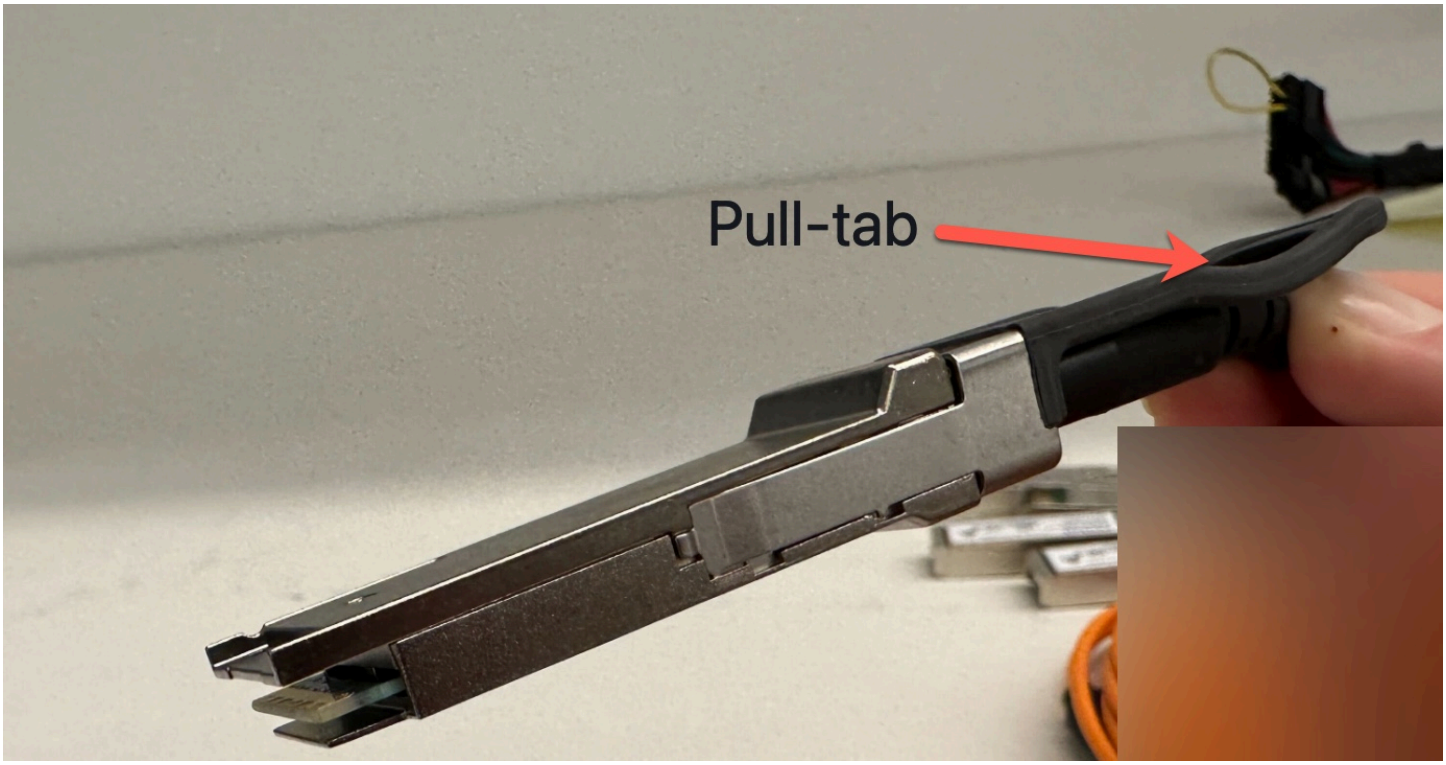


**Note**

AWS Outposts servers have a physical RJ45 port beside the QSFP port. However, this RJ45 port is not enabled for any customer use. If you require RJ45 1GbE connectivity, use the included QSFP cable to connect a 10GBASE-X SFP+ to a 1GbE RJ45 media converter.

One end of the QSFP cable has a single connector. Connect this end to the server.

The following image shows the end of the cable with the single connector:




The other end of the QSFP cable has 4 breakout cables labeled 1 through 4. Use the cable labeled 1 for LNI link traffic and the cable labeled 2 for service link traffic.

The following image shows the end of the cable with the 4 breakout cables:


**To connect the server to the network with the QSFP breakout cable**

1. Locate the QSFP breakout cable that came with the server.
2. Connect the single end of the QSFP breakout cable to the QSFP port on the server.

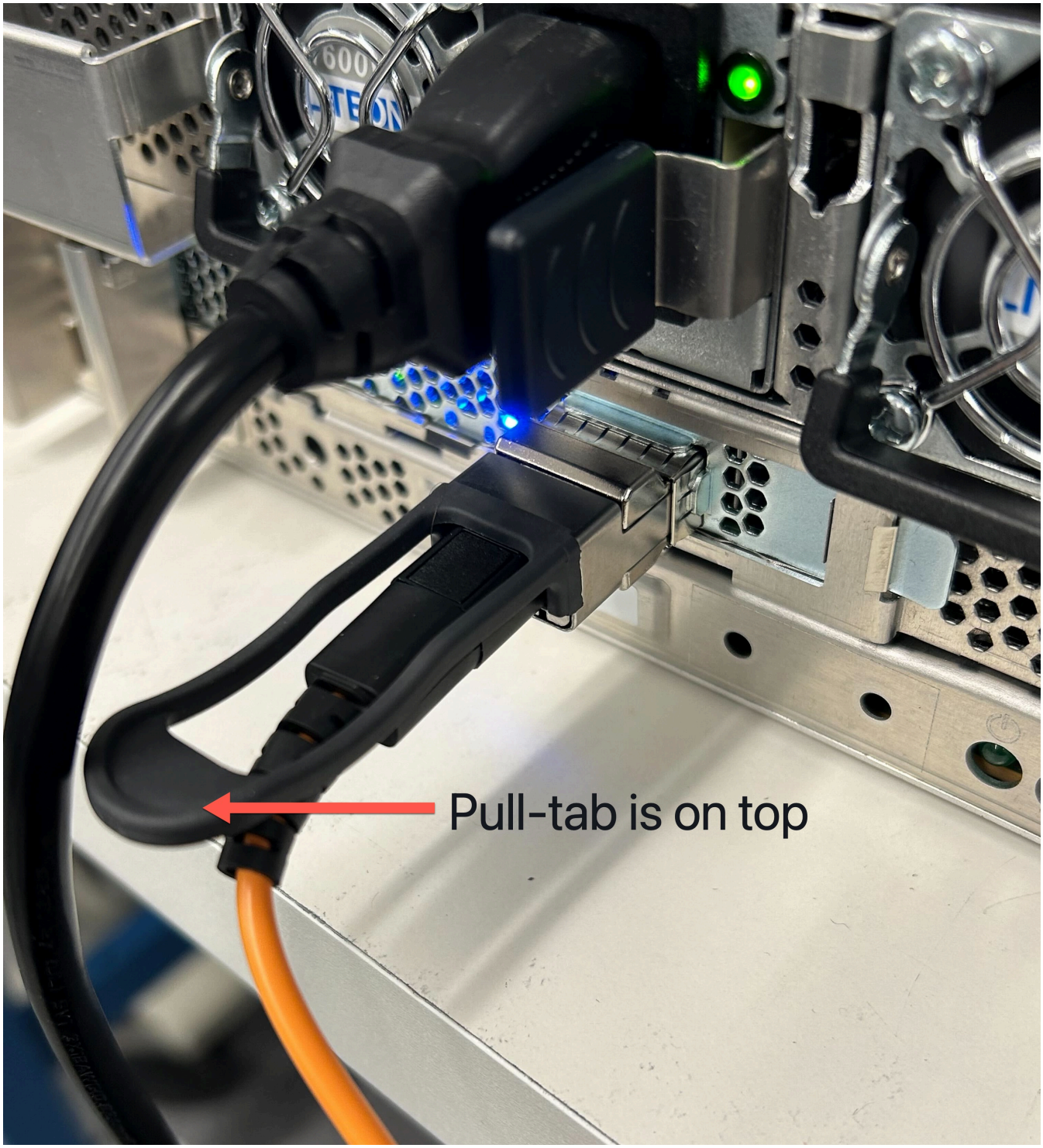
    1. Locate the QSFP port.

        The following image shows the location of the QSFP port on the 2U server.

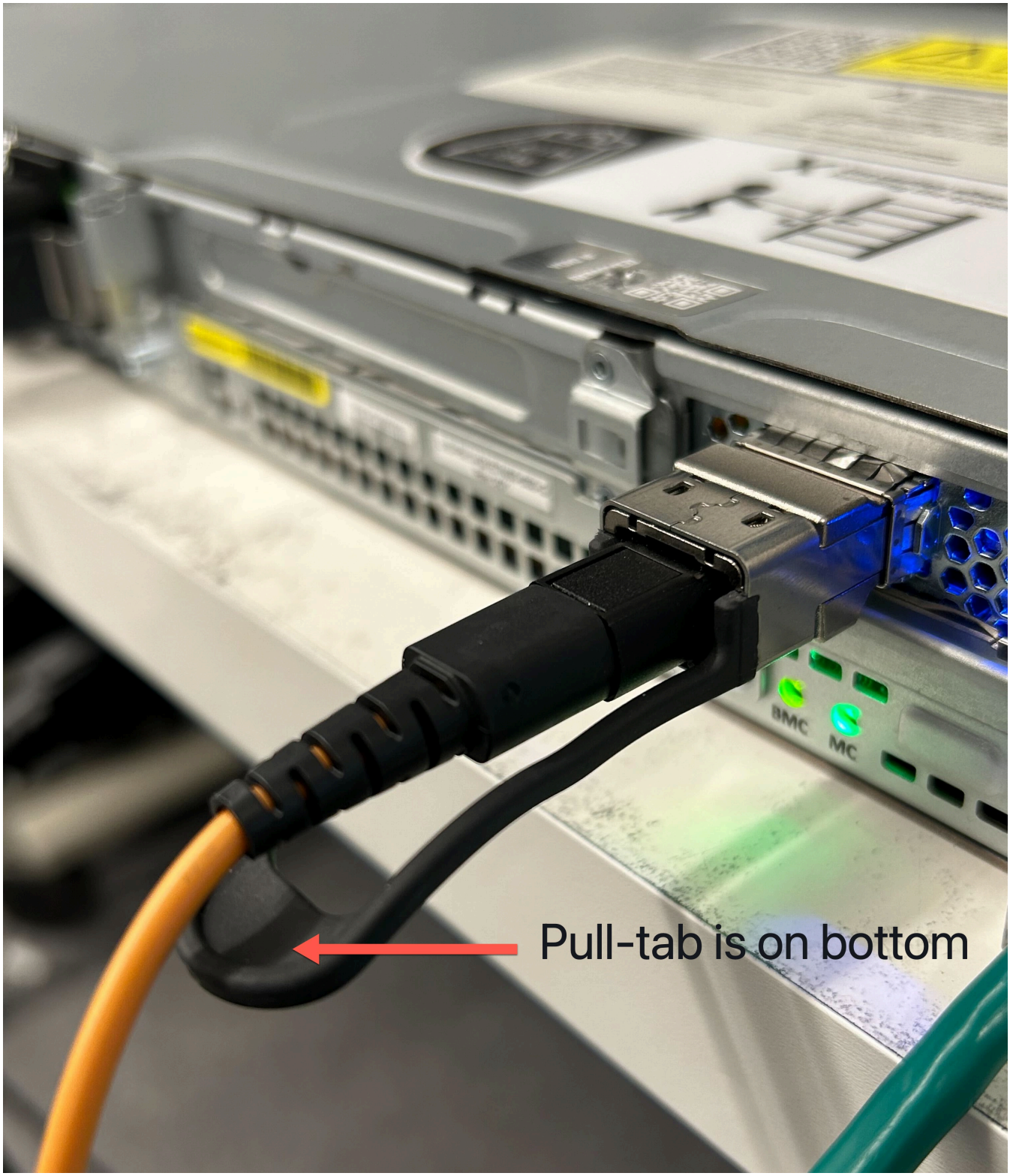

The following image shows the location of the QSFP port on the 1U server.


2. Plug in the QSFP with the pull-tab in the correct orientation.

   For the 2U server, plug in the QSFP with the pull-tab on top as the following image shows.




For the 1U server, plug in the QSFP with the pull-tab on the bottom as the following image shows.




3. Ensure that you feel or hear a click when you plug the cables in. This indicates that you plugged in the cables correctly.

3. Connect breakouts 1 and 2 of the QSFP cable to the upstream networking device.

> **Important**
>
> Both of the following cables are required for an Outpost server to function.
>
> - Use the cable labeled 1 for LNI link traffic.
> - Use the cable labeled 2 for service link traffic.

## Step 6: Authorize server

To authorize the server, you must connect your laptop to the server with a USB cable, then use a command-based serial protocol to test the connection and authorize the server. In addition to IAM credentials, you need a USB cable, a laptop, and serial terminal software, such as PuTTY or **screen**, to complete these steps.

Alternatively, if you have an Android phone or tablet with a USB-C or micro-USB connector with USB On The Go (OTG) support, you can use the **Outposts Server Activator** app to walk you through the server-authorization process. You can download the app from Google Play

Consider the following information about authorizing the server:

- To authorize the server, you or the party installing the server needs IAM credentials in the AWS account that contains the Outpost. For more information, see the section called “Step 1: Grant permissions”.
- You do not need to authenticate with the IAM credentials to test your connection.
- Consider testing the connection before you use the export command to set IAM credentials as environment variables.
- To protect your account, Outpost Configuration Tool never saves your IAM credentials.
- To connect your laptop to the server, always plug the USB cable into your laptop first and then into the server.

### Tasks

- Connect your laptop to the server
- Create a serial connection to the server
- Test the connection

- Authorize the server
- Verify the NSK LEDs

## Connect your laptop to the server

Connect the USB cable to your laptop first and then to the server. The server includes a USB chip that creates a virtual serial port available to you on the laptop. You can use this virtual serial port to connect to the server with serial terminal emulation software. You can only use this virtual serial port to run Outpost Configuration Tool commands.

### To connect the laptop to the server

Plug the USB cable into your laptop first, then into the server.

> **Note**
>
> The USB chip requires drivers to create the virtual serial port. Your operating system should automatically install the required drivers if they are not already present. To download and install the drivers, see Installation Guides from FTDI.

## Create a serial connection to the server

This section contains instructions for using popular serial terminal programs, but you are not required to use these programs. Use the serial terminal program you prefer with a connection speed of `115200` baud.

### Examples

- Windows serial connection
- Mac serial connection

### Windows serial connection

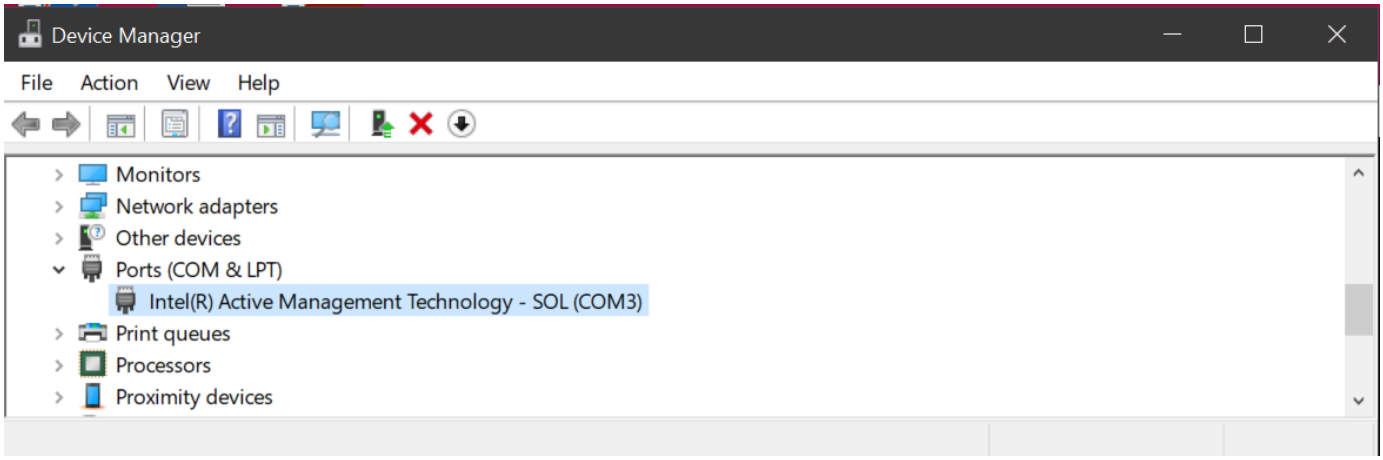
The following instructions are for PuTTY on Windows. PuTTY is free, but you may have to download it.

### Download PuTTY

Download and install PuTTY from the PuTTY download page.

**To create a serial terminal on Windows using PuTTY**

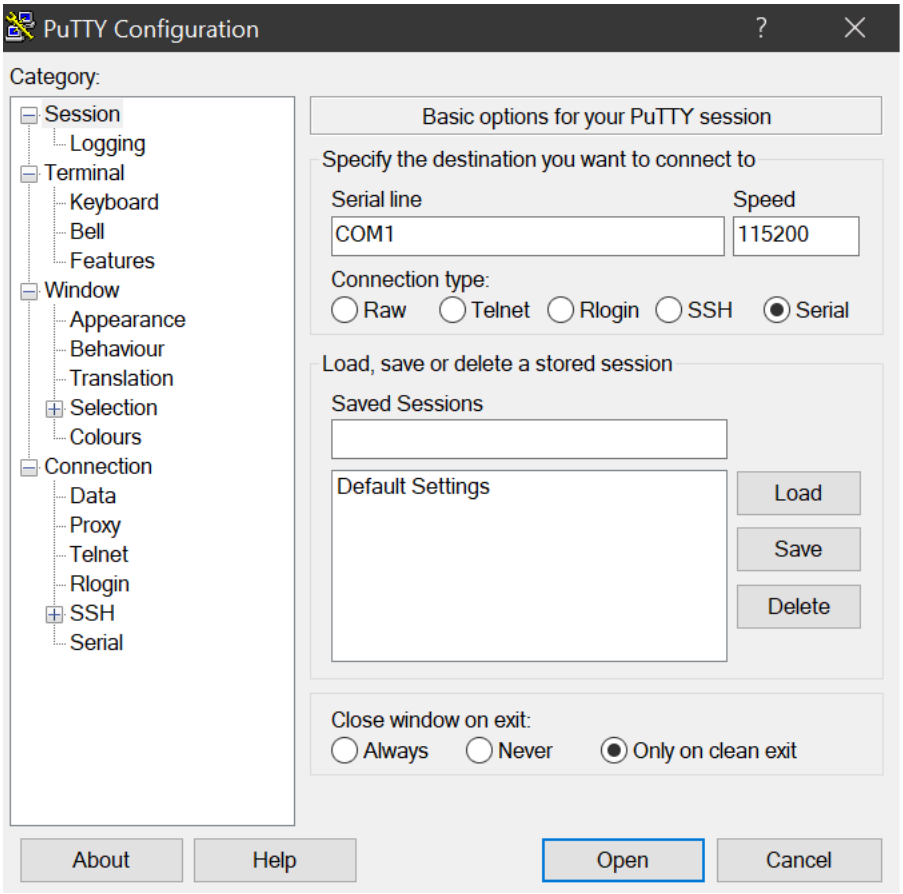
1. Plug the USB cable into your Windows laptop first, then into the server.
2. From the Desktop, right-click **Start**, and choose **Device Manager**.
3. In **Device Manager**, expand **Ports (COM & LPT)** to determine the COM port for the USB serial connection. You will see a node named **USB Serial Port (COM#)**. The value for the COM port depends on your hardware.




4. In PuTTY, from **Session**, choose **Serial** for **Connection type**, and then enter the following information:

   - Under **Serial line**, enter the COM# port from Device Manager.
   - Under **Speed**, enter: `115200`

   The following image shows an example on the **PuTTY Configuration** page:




5. Choose **Open**.

   An empty console window appears. It can take between 1 to 2 minutes for one of the following to appear:

   - `Please wait for the system to stabilize. This can take up to 900 seconds, so far` *`x seconds`* `have elapsed on this boot.`
   - The `Outpost>` prompt.

### Mac serial connection

The following instructions are for **screen** on macOS. You can find **screen** included with the operating system.

#### To create a serial terminal on macOS using screen

1. Plug the USB cable into your Mac laptop first, then into the server.
2. In Terminal, list `/dev` with a `*usb*` filter for output to find the virtual serial port.

```
ls -ltr /dev/*usb*
```

The serial device appears as `tty`. For example, consider the following sample output from the previous list command:

```
ls -ltr /dev/*usb*
crw-rw-rw-  1 root  wheel   21,   3 Feb  8 15:48 /dev/cu.usbserial-EXAMPLE1
crw-rw-rw-  1 root  wheel   21,   2 Feb  9 08:56 /dev/tty.usbserial-EXAMPLE1
```

3. In Terminal, use **screen** with the serial device and a baud rate of the serial connection to set up the serial connection. In the following command, replace *EXAMPLE1* with the value from your laptop.

```
screen /dev/tty.usbserial-EXAMPLE1 115200
```

An empty console window appears. It can take between 1 to 2 minutes for one of the following to appear:

- `Please wait for the system to stabilize. This can take up to 900 seconds, so far x seconds have elapsed on this boot.`
- The `Outpost>` prompt.

## Test the connection

This section describes how to use Outpost Configuration Tool to test the connection. You don't need IAM credentials to test the connection. Your connection needs to be able to resolve DNS to access the AWS Region.

1. Test the links and gather information about the connection
2. Test for DNS resolver
3. Test for access to the AWS Region

### To test the links

1. Plug the USB cable into your laptop first and then into the server.

2. Use a serial terminal program, such as PuTTY or **screen**, to connect to the server. For more information, see the section called “Create a serial connection to the server”.
3. Press **Enter** to access the Outpost Configuration Tool command prompt.

```
Outpost>
```

> **Note**
>
> If you see a persistent red light inside the chassis of the server on the left-hand side after you power on and you cannot connect to Outpost Configuration Tool, you may need to power down and drain the server to proceed. To drain the server, disconnect all network and power cables, wait five minutes, then power up and connect the network again.

4. Use **describe-links** to return information about the network links on the server. Outpost servers must have one service link and one local network interface (LNI) link.

```
Outpost>describe-links
---
service_link_connected: True
local_link_connected: False
links:
-
  name: local_link
  connected: False
  mac: 00:00:00:00:00:00
-
  name: service_link
  connected: True
  mac: 0A:DC:FE:D7:8E:1F
checksum: 0x46FDC542
```

If you get `connected: False` for either link, troubleshoot the network connection on the hardware.

5. Use **describe-ip** to return the IP assignment status and configuration of the service link.

```
Outpost>describe-ip
---
links:
```

```
-
  name: service_link
  configured: True
  ip: 192.168.0.0
  netmask: 255.255.0.0
  gateway: 192.168.1.1
  dns: [ "192.168.1.1" ]
  ntp: [ ]
checksum: 0x8411B47C
```

The NTP value might be missing as NTP is optional in a DHCP option set. You should have no other missing values.

### To test for DNS

1. Plug the USB cable into your laptop first and then into the server.
2. Use a serial terminal program, such as PuTTY or **screen**, to connect to the server. For more information, see the section called “Create a serial connection to the server”.
3. Press **Enter** to access the Outpost Configuration Tool command prompt.

```
Outpost>
```

> **Note**
>
> If you see a persistent red light inside the chassis of the server on the left-hand side after you power on and you cannot connect to Outpost Configuration Tool, you may need to power down and drain the server to proceed. To drain the server, disconnect all network and power cables, wait five minutes, then power up and connect the network again.

4. Use **export** to enter the parent Region of the Outpost server as the value for `AWS_DEFAULT_REGION`.

   `AWS_DEFAULT_REGION=Region`

```
Outpost>export AWS_DEFAULT_REGION=us-west-2

result: OK
```

```
checksum: 0xB2A945RE
```

- Do not include a space before or after the equal (=) sign.
- No environment values are saved. You must export AWS Region each time you run Outpost Configuration Tool.
- If you are using a third-party to install the server, you must provide the third-party with the parent Region.

5. Use **describe-resolve** to determine if the Outpost server can reach a DNS resolver and resolve the IP address of the Outpost configuration endpoint in the Region. Requires at least one link with an IP configuration.

```
Outpost>describe-resolve
---
dns_responding: True
dns_resolving: True
dns: [ "198.xx.xxx.xx", "198.xx.xxx.xx" ]
query: outposts.us-west-2.amazonaws.com
records: [ "18.xxx.xx.xxx", "44.xxx.xxx.xxx", "44.xxx.xxx.xxx" ]
checksum: 0xB6A961CE
```

**To test access to AWS Regions**

1. Plug the USB cable into your laptop first and then into the server.
2. Use a serial terminal program, such as PuTTY or **screen**, to connect to the server. For more information, see the section called “Create a serial connection to the server”.
3. Press **Enter** to access the Outpost Configuration Tool command prompt.

```
Outpost>
```

> **Note**
>
> If you see a persistent red light inside the chassis of the server on the left-hand side after you power on and you cannot connect to Outpost Configuration Tool, you may need to power down and drain the server to proceed. To drain the server, disconnect all network and power cables, wait five minutes, then power up and connect the network again.

4. Use **export** to enter the parent Region of the Outpost server as the value for `AWS_DEFAULT_REGION`.

   `AWS_DEFAULT_REGION=`*`Region`*

   ```
   Outpost>export AWS_DEFAULT_REGION=us-west-2

   result: OK
   checksum: 0xB2A945RE
   ```

   - Do not include a space before or after the equal (=) sign.
   - No environment values are saved. You must export AWS Region each time you run Outpost Configuration Tool.
   - If you are using a third-party to install the server, you must provide the third-party with the parent Region.

5. Use **describe-reachability** to determine if the Outpost server can reach the Outpost configuration endpoint in the Region. Requires a working DNS configuration, which you can determine by using **describe-resolve**.

   ```
   Outpost>describe-reachability
   ---
   is_reachable: True
   src_ip: 10.0.0.0
   dst_ip: 54.xx.x.xx
   dst_port: xxx
   checksum: 0xCB506615
   ```

   - `is_reachable` indicates the outcome of the test
   - `src_ip` is the IP address of the server
   - `dst_ip` is the IP address of the Outpost configuration endpoint in the Region
   - `dst_port` is the port the server used to connect to `dst_ip`

## Authorize the server

This section describes how to use Outpost Configuration Tool and the IAM credentials from the AWS account that contains the Outpost to authorize the server.

**To authorize the server**

1. Plug the USB cable into your laptop first and then into the server.
2. Use a serial terminal program, such as PuTTY or **screen**, to connect to the server. For more information, see the section called “Create a serial connection to the server”.
3. Press **Enter** to access the Outpost Configuration Tool command prompt.

   ```
   Outpost>
   ```

   > **Note**
   >
   > If you see a persistent red light inside the chassis of the server on the left-hand side after you power on and you cannot connect to Outpost Configuration Tool, you may need to power down and drain the server to proceed. To drain the server, disconnect all network and power cables, wait five minutes, then power up and connect the network again.

4. Use **export** to enter your IAM credentials into Outpost Configuration Tool. If you are using a third-party to install the server, you must provide the third-party with the IAM credentials.

   To authenticate, you must export the following four variables. Export one variable at a time. Do not include a space before or after the equal (=) sign.

   - `AWS_ACCESS_KEY_ID=`*`access-key-id`*
   - `AWS_SECRET_ACCESS_KEY=`*`secret-access-key`*
   - `AWS_SESSION_TOKEN=`*`session-token`*
     - Use the AWS CLI `GetSessionToken` command to get the `AWS_SESSION_TOKEN`. For more information, see get-session-token in the *AWS CLI Command Reference*.

       > **Note**
       >
       > You must have the AWSOutpostsAuthorizeServerPolicy attached to your IAM role to get the `AWS_SESSION_TOKEN`.

     - To install the AWS CLI, see Installing or updating the latest version of the AWS CLI in the *AWS CLI User Guide for Verrsion 2*.
   - `AWS_DEFAULT_REGION=`*`Region`*

Use the parent Region of the Outpost server as the value for `AWS_DEFAULT_REGION`. If you are using a third party to install the server, you must provide the third party with the parent Region.

The output in the following examples show successful exports.

```
Outpost>export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE

result: OK
checksum: example-checksum
```

```
Outpost>export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

result: OK
checksum: example-checksum
```

```
Outpost>export AWS_SESSION_TOKEN=MIICiTCCAfICCQD6m7oRw0uXOjANBgk
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=

result: OK
checksum: example-checksum
```

```
Outpost>export AWS_DEFAULT_REGION=us-west-2

result: OK
checksum: example-checksum
```

5. Use **start-connection** to create a secure connection to the Region.

   The output in the following example shows a connection successfully started.

```
Outpost>start-connection

is_started: True
asset_id: example-asset-id
connection_id: example-connection-id
timestamp: 2021-10-01T23:30:26Z
checksum: example-checksum
```

6. Wait for about 5 minutes.
7. Use **get-connection** to check if the connection to the Region has been established.

   The output in the following example shows a successful connection.

```
Outpost>get-connection


---
keys_exchanged: True
connection_established: True
exchange_active: False
primary_peer: xx.xx.xx.xx:xxx
primary_status: success
primary_connection_id: a1b2c3d4567890abcdefEXAMPLE11111
primary_handshake_age: 1111111111
primary_server_public_key: AKIAIOSFODNN7EXAMPLE
primary_client_public_key: AKIAI44QH8DHBEXAMPLE
primary_server_endpoint: xx.xx.xx.xx:xxx
secondary_peer: xx.xxx.xx.xxx:xxx
secondary_status: success
secondary_connection_id: a1b2c3d4567890abcdefEXAMPLE22222
secondary_handshake_age: 1111111111
secondary_server_public_key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
secondary_client_public_key: je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
secondary_server_endpoint: xx.xxx.xx.xxx:xxx
timestamp: 2023-02-22T22:19:28Z
checksum: 0x83FA0123
```

After `keys_exchanged` and `connection_established` changes to `True`, the Outpost server is automatically provisioned and updated to the latest software and configuration.

**Note**

Note the following about the provisioning process:

- After activation completes, it can take up to 10 hours until your Outpost server is usable.
- You must keep the Outpost server's power and network connected and stable during this process.
- It is normal for the service link to fluctuate during this process.
- If `exchange_active` is `True`, the connection is still establishing. Retry in 5 minutes.
- If `keys_exchanged` or `connection_established` is `False`, and if `exchange_active` is `True`, the connection is still establishing. Retry in 5 minutes.
- If `keys_exchanged` or `connection_established` is `False` even after 1 hour, contact AWS Support Center.
- If the message `primary_status: No such asset id found.` appears, confirm the following:
  - You specified the correct Region.
  - You are using the same account as the one used to order the Outpost server.

  If the Region is correct and you are using the same account as the one used to order the Outpost server, contact AWS Support Center.
- The `LifeCycleStatus` attribute of the Outpost will transition from `Provisioning` to `Active`. You will then receive an email letting you know that your Outpost server is provisioned and activated.
- You don't need to re-authorize the Outposts server after the Outposts server is activated.

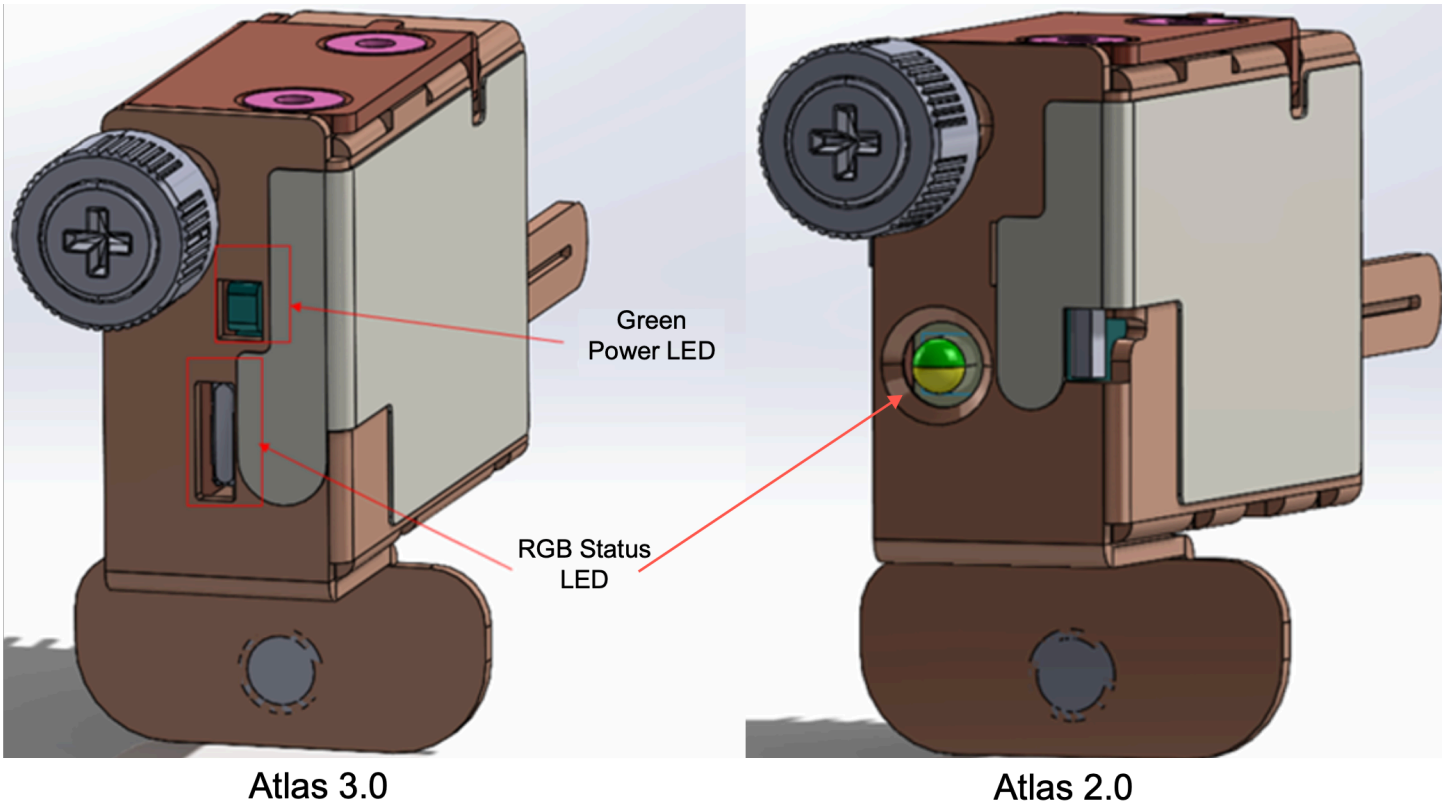
8. After you make a successful connection, you can disconnect your laptop from the server.

## Verify the NSK LEDs

After the provisioning process completes, check the NSK LEDs.

AWS Outposts supports two versions of NSK: Atlas 2.0 and Atlas 3.0. Both NSK versions have a RGB **Status** LED. In addition, the Atlas 3.0 has a green **Power** LED.

The following image shows the location of the LEDs on the Atlas 2.0 and Atlas 3.0:




**To verify the Status and Power LEDs on the NSK**

1. Check the color of the RGB Status LED. If the color is green, the NSK is healthy. If the color is not green, contact AWS Support.
2. If you have an Atlas 3.0 NSK, check the green Power LED. If the green light is on, the NSK is correctly connected to the host and has power. If the green light is not on, contact AWS Support.

## Outpost Configuration Tool command reference

The Outpost Configuration Tool provides the following commands.

### Commands

- Export
- Echo

- Describe links
- Describe IP
- Describe resolve
- Describe reachability
- Start connection
- Get connection

## Export

### export

Use **export** to set IAM credentials as environment variables.

### Syntax

```
Outpost>export variable=value
```

**export** takes the variable assignment statement.

Must use the following format: *variable*=*value*

To authenticate, you must export the following four variables. Export one variable at a time. Do not include a space before or after the equal (=) sign.

- AWS_ACCESS_KEY_ID=*access-key-id*
- AWS_SECRET_ACCESS_KEY=*secret-access-key*
- AWS_SESSION_TOKEN=*session-token*
- AWS_DEFAULT_REGION=*Region*

  Use the parent Region of the Outpost server as the value for AWS_DEFAULT_REGION.

### Example : successful credential imports

```
Outpost>export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE

result: OK
checksum: example-checksum
```

```
Outpost>export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

result: OK
checksum: example-checksum
```

```
Outpost>export AWS_SESSION_TOKEN=MIICiTCCAfICCQD6m7oRw0uXOjANBgk
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=

result: OK
checksum: example-checksum
```

```
Outpost>export AWS_DEFAULT_REGION=us-west-2

result: OK
checksum: example-checksum
```

## Echo

### echo

Use **echo** to display the value that you set for a variable using the **export** command.

### Syntax

```
Outpost>echo $variable-name
```

The *variable-name* can be one of the following:

- AWS_ACCESS_KEY_ID
- AWS_SECRET_ACCESS_KEY

- AWS_SESSION_TOKEN
- AWS_DEFAULT_REGION

### Example : Success

```
Outpost>export AWS_DEFAULT_REGION=us-west-2

result: OK
checksum: example-checksum

---

Outpost>echo $AWS_DEFAULT_REGION

variable name: AWS_DEFAULT_REGION
variable value: us-west-2
checksum: example-checksum
```

### Example : Failure because the variable value was not set with the export command

```
Outpost> echo $AWS_ACCESS_KEY_ID

error_type: execution_error
error_attributes:
  AWS_ACCESS_KEY_ID: no value set
error_message: No value set for AWS_ACCESS_KEY_ID using export.
checksum: example-checksum
```

### Example : Failure because the variable name is not valid

```
Oupost>echo $foo

error_type: invalid_argument
error_attributes:
  foo: invalid variable name
error_message: Variables can only be AWS credentials.
checksum: example-checksum
```

### Example : Failure because of a syntax issue

```
Outpost>echo AWS_SECRET_ACCESS_KEY
```

```
error_type: invalid_argument
error_attributes:
  AWS_SECRET_ACCESS_KEY: not a variable
error_message: Expecting $ before variable name.
checksum: example-checksum
```

## Describe links

### describe-links

Use **describe-links** to return information about the network links on the server. Outpost servers must have one service link and one local network interface (LNI) link.

**Syntax**

```
Outpost>describe-links
```

**describe-links** takes no arguments.

## Describe IP

### describe-ip

Use **describe-ip** to return the IP assignment status and configuration of each connected link.

**Syntax**

```
Outpost>describe-ip
```

**describe-ip** takes no arguments.

## Describe resolve

### describe-resolve

Use **describe-resolve** to determine if the Outpost server can reach a DNS resolver and resolve the IP address of the Outpost configuration endpoint in the Region. Requires at least one link with an IP configuration.

**Syntax**

```
Outpost>describe-resolve
```

**describe-resolve** takes no arguments.

## Describe reachability

### describe-reachability

Use **describe-reachability** to determine if the Outpost server can reach the Outpost configuration endpoint in the Region. Requires a working DNS configuration, which you can determine by using **describe-resolve**.

**Syntax**

```
Outpost>describe-reachability
```

**describe-reachability** takes no arguments.

## Start connection

### start-connection

Use **start-connection** to initiate a connection with the Outpost service in the Region. This command sources the Signature Version 4 (SigV4) credentials from the environment variables you loaded with **export**. The connection runs asynchronously and returns immediately. To check the status of the connection, use **get-connection**.

**Syntax**

```
Outpost>start-connection [0|1]
```

**start-connection** takes an optional connection index to initiate another connection. Only values of `0` and `1` are valid.

### Example : connection started

```
Outpost>start-connection
```

```
is_started: True
asset_id: example-asset-id
connection_id: example-connecdtion-id
timestamp: 2021-10-01T23:30:26Z
checksum: example-checksum
```

## Get connection

### get-connection

Use **get-connection** to return the status of the connection.

### Syntax

```
Outpost>get-connection [0|1]
```

**get-connection** takes an optional connection index to return the status of another connection. Only values of 0 and 1 are valid.

### Example : successful connection

```
Outpost>get-connection


---
keys_exchanged: True
connection_established: True
exchange_active: False
primary_peer: xx.xx.xx.xx:xxx
primary_status: success
primary_connection_id: a1b2c3d4567890abcdefEXAMPLE11111
primary_handshake_age: 1111111111
primary_server_public_key: AKIAIOSFODNN7EXAMPLE
primary_client_public_key: AKIAI44QH8DHBEXAMPLE
primary_server_endpoint: xx.xx.xx.xx:xxx
secondary_peer: xx.xxx.xx.xxx:xxx
secondary_status: success
secondary_connection_id: a1b2c3d4567890abcdefEXAMPLE22222
secondary_handshake_age: 1111111111
secondary_server_public_key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
secondary_client_public_key: je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
```

```
secondary_server_endpoint: xx.xxx.xx.xxx:xxx
timestamp: 2023-02-22T22:19:28Z
checksum: 0x83FA0123
```

Note:

- If `exchange_active` is `True`, the connection is still establishing. Retry in 5 minutes.
- If `keys_exchanged` or `connection_established` is `False`, and if `exchange_active` is `True`, the connection is still establishing. Retry in 5 minutes.

If the issue persists after 1 hour, contact AWS Support Center.

# Launch an instance on your Outpost server

After your Outpost is installed and the compute and storage capacity is available for use, you can get started by creating resources. For example, you can launch Amazon EC2 instances.

**Prerequisite**

You must have an Outpost installed at your site. For more information, see Create an Outpost and order Outpost capacity.

**Tasks**

- Step 1: Create a subnet
- Step 2: Launch an instance on the Outpost
- Step 3: Configure connectivity
- Step 4: Test the connectivity

## Step 1: Create a subnet

You can add Outpost subnets to any VPC in the AWS Region for the Outpost. When you do so, the VPC also spans the Outpost. For more information, see Network components.

> **Note**
>
> If you are launching an instance in an Outpost subnet that has been shared with you by another AWS account, skip to Step 2: Launch an instance on the Outpost.

**To create an outpost subnet**

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions**, **Create subnet**. You are redirected to create a subnet in the Amazon VPC console. We select the Outpost for you and the Availability Zone that the Outpost is homed to.
4. Select a VPC and specify an IP address range for the subnet.
5. Choose **Create**.
6. After the subnet is created, enable the subnet for local network interfaces.

## Step 2: Launch an instance on the Outpost

You can launch EC2 instances in the Outpost subnet that you created, or in an Outpost subnet that has been shared with you. Security groups control inbound and outbound VPC traffic for instances in an Outpost subnet, just as they do for instances in an Availability Zone subnet. To connect to an EC2 instance in an Outpost subnet, you can specify a key pair when you launch the instance, just as you do for instances in an Availability Zone subnet.

**Considerations**

- Instances on Outposts servers include instance store volumes but not EBS volumes. Choose an instance size with enough instance storage to meet the needs of your application. For more information, see Instance store volumes in the *Amazon EC2 User Guide*.
- You must specify an AMI with only a single snapshot. AMIs with more than one snapshot are not supported.
- The data on instance store volumes persists after an instance reboot but does not persist after instance termination. To retain the long-term data on your instance store volumes beyond the lifetime of the instance, be sure to back up the data to persistent storage, such as an Amazon S3 bucket or a network storage device in your on-premises network.
- To connect an instance in an Outpost subnet to your on-premises network, you must add a local network interface, as described in the following procedure.

**To launch instances in your Outpost subnet**

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.

2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions, View details**.
4. On the **Outpost summary** page, choose **Launch instance**. You are redirected to the instance launch wizard in the Amazon EC2 console. We select the Outpost subnet for you, and show you only the instance types that are supported by your Outposts servers.
5. Choose an instance type that is supported by your Outposts servers.
6. (Optional) You can add a local network interface now or after you create the instance. To add it now, expand **Advanced network configuration** and choose **Add network interface**. Choose the Outpost subnet. This creates a network interface for the instance using device index 1. If you specified 1 as the LNI device index for the Outpost subnet, then this network interface will be the local network interface for the instance.
7. Complete the wizard to launch the instance in your Outpost subnet. For more information, see the following in the *Amazon EC2 User Guide*:

   - Linux – Launch an instance using the new launch instance wizard
   - Windows – Launch an instance using the new launch instance wizard

## Step 3: Configure connectivity

If you did not add a local network interface to your instance during instance launch, you must do so now. For more information, see Add an LNI after launch.

You must configure the local network interface for the instance with an IP address from your local network. Typically, you do this by using DHCP. For information, see the documentation for the operating system running on the instance. Search for information about configuring additional network interfaces and secondary IP addresses.

## Step 4: Test the connectivity

You can test connectivity by using the appropriate use cases.

**Test connectivity from your local network to the Outpost**

From a computer in your local network, run the `ping` command to the Outpost instance's local network interface IP address.

```
ping 10.0.3.128
```

The following is example output.

```
Pinging 10.0.3.128

Reply from 10.0.3.128:  bytes=32 time=<1ms TTL=128
Reply from 10.0.3.128:  bytes=32 time=<1ms TTL=128
Reply from 10.0.3.128:  bytes=32 time=<1ms TTL=128

Ping statistics for 10.0.3.128
Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

Approximate round trip time in milliseconds
Minimum = 0ms,  Maximum = 0ms,  Average = 0ms
```

**Test the connectivity from an Outpost instance to your local network**

Depending on your operating system, use **ssh** or **rdp** to connect to the private IP address of your Outpost instance. For information about connecting to a Linux instance, see Connect to your Linux instance in the *Amazon EC2 User Guide for Linux Instances*. For information about connecting to a Windows instance, see Connect to your Windows instance in the *Amazon EC2 User Guide for Windows Instances*.

After the instance is running, run the `ping` command to an IP address of a computer in your local network. In the following example, the IP address is 172.16.0.130.

```
ping 172.16.0.130
```

The following is example output.

```
Pinging 172.16.0.130

Reply from 172.16.0.130:  bytes=32 time=<1ms TTL=128
Reply from 172.16.0.130:  bytes=32 time=<1ms TTL=128
Reply from 172.16.0.130:  bytes=32 time=<1ms TTL=128

Ping statistics for 172.16.0.130
Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

Approximate round trip time in milliseconds
Minimum = 0ms,  Maximum = 0ms,  Average = 0ms
```

### Test connectivity between the AWS Region and the Outpost

Launch an instance in the subnet in the AWS Region. For example, use the run-instances command.

```
aws ec2 run-instances \
    --image-id ami-abcdefghi1234567898 \
    --instance-type c5.large \
    --key-name MyKeyPair \
    --security-group-ids sg-1a2b3c4d123456787 \
    --subnet-id subnet-6e7f829e123445678
```

After the instance is running, perform the following operations:

1. Get the private IP address of the instance in the AWS Region. This information is available in the Amazon EC2 console on the instance detail page.
2. Depending on your operating system, use **ssh** or **rdp** to connect to the private IP address of your Outpost instance.
3. Run the **ping** command from your Outpost instance, specifying the IP address of the instance in the AWS Region.

   ```
   ping 10.0.1.5
   ```

   The following is example output.

   ```
   Pinging 10.0.1.5

   Reply from 10.0.1.5:  bytes=32 time=<1ms TTL=128
   Reply from 10.0.1.5:  bytes=32 time=<1ms TTL=128
   Reply from 10.0.1.5:  bytes=32 time=<1ms TTL=128

   Ping statistics for 10.0.1.5
   Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

   Approximate round trip time in milliseconds
   Minimum = 0ms,  Maximum = 0ms,  Average = 0ms
   ```

# AWS Outposts connectivity to AWS Regions

AWS Outposts supports wide area network (WAN) connectivity through the service link connection.

> **Note**
>
> You cannot use private connectivity for your service link connection that connects your Outpost server to your AWS Region or AWS Outposts home Region.

**Contents**



## Connectivity through service links

During AWS Outposts provisioning, you or AWS creates a service link connection that connects your Outpost back to your chosen AWS Region or AWS Outposts home Region. The service link is an encrypted set of VPN connections that are used whenever the Outpost communicates with your chosen home Region. You use a virtual LAN (VLAN) to segment traffic on the service link. The service link VLAN enables communication between the Outpost and the AWS Region for both management of the Outpost and intra-VPC traffic between the AWS Region and Outpost.

The Outpost is able to create the service link VPN back to the AWS Region through public Region connectivity. To do so, the Outpost needs connectivity to the AWS Region's public IP ranges, either through the public internet or AWS Direct Connect public virtual interface. This connectivity can be through specific routes in the service link VLAN, or through a default route of 0.0.0.0/0. For more information about the public ranges for AWS, see AWS IP Address Ranges.

After the service link is established, the Outpost is in service and managed by AWS. The service link is used for the following traffic:

- Management traffic to the Outpost through the service link, including internal control plane traffic, internal resource monitoring, and updates to firmware and software.
- Traffic between the Outpost and any associated VPCs, including customer data plane traffic.

## Service link maximum transmission unit (MTU) requirements

The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The network must support 1500-bytes MTU between the Outpost and the service link endpoints in the parent AWS Region. For information on the required MTU between an instance in the Outpost and an instance in the AWS Region through the service link, see Network maximum transmission unit (MTU) for your Amazon EC2 instance in the *Amazon EC2 User Guide for Linux Instances*.

## Service link bandwidth recommendations

For an optimal experience and resiliency, AWS recommends that you use redundant connectivity of at least 500 Mbps for the service link connection to the AWS Region. The maximum utilization for each Outpost server is 500 Mbps. To increase the connection speed, use multiple Outpost servers. For example, if you have three AWS Outposts servers, the maximum connection speed increases to 1.5 Gbps (1,500 Mbps). For more information, see Service link traffic for servers.

Your AWS Outposts service link bandwidth requirements vary depending on workload characteristics, such as AMI size, application elasticity, burst speed needs, and Amazon VPC traffic to the Region. Note that AWS Outposts servers do not cache AMIs. AMIs are downloaded from the Region with every instance launch.

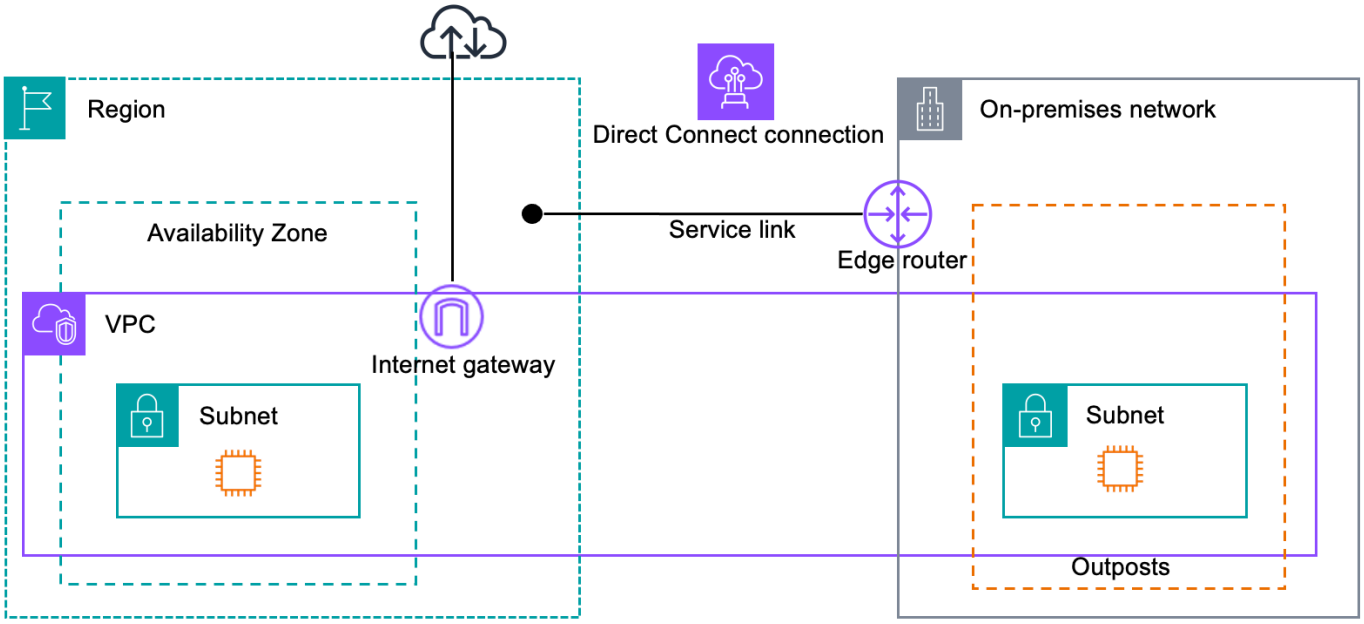
To receive a custom recommendation about the service link bandwidth required for your needs, contact your AWS sales representative or APN partner.

## Firewalls and the service link

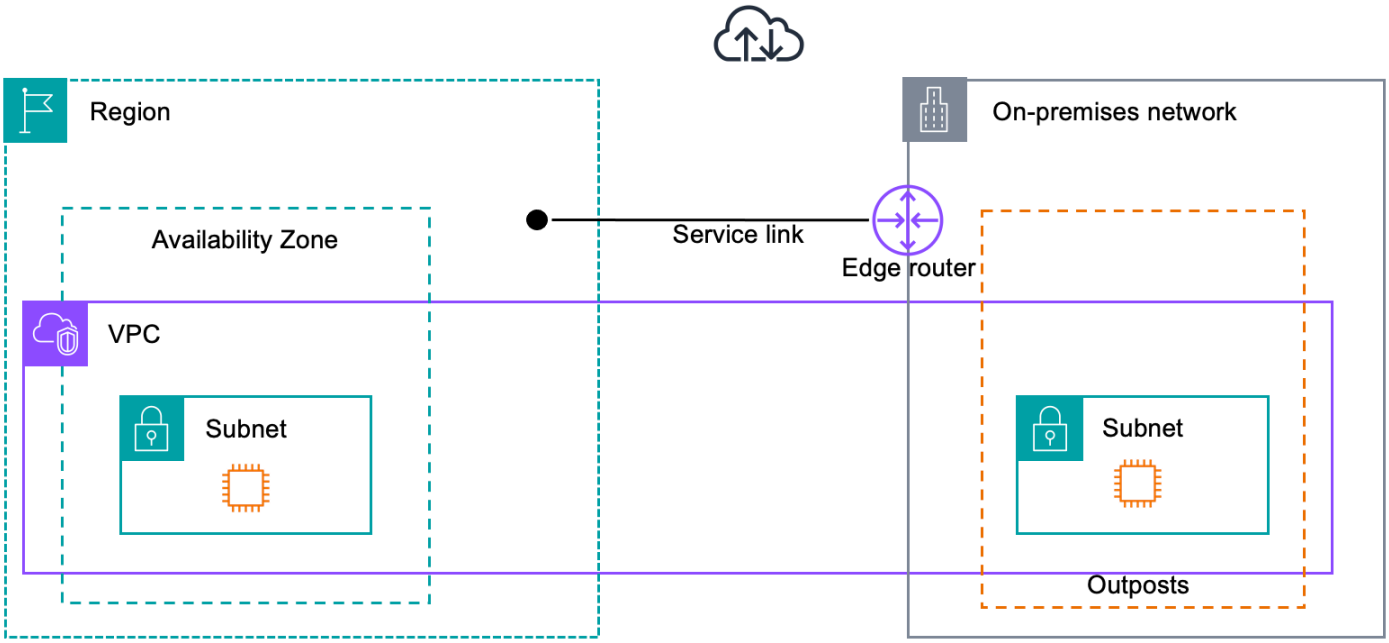
This section discusses firewall configurations and the service link connection.

In the following diagram, the configuration extends the Amazon VPC from the AWS Region to the Outpost. An AWS Direct Connect public virtual interface is the service link connection. The following traffic goes over the service link and the AWS Direct Connect connection:

- Management traffic to the Outpost through the service link
- Traffic between the Outpost and any associated VPCs




If you are using a stateful firewall with your internet connection to limit connectivity from the public internet to the service link VLAN, you can block all inbound connections that initiate from the internet. This is because the service link VPN initiates only from the Outpost to the Region, not from the Region to the Outpost.




If you use a firewall to limit the connectivity from the service link VLAN, you can block all inbound connections. You must allow outbound connections back to the Outpost from the AWS Region as

per the following table. If the firewall is stateful, outbound connections from the Outpost that are allowed, meaning that they were initiated from the Outpost, should be allowed back inbound.

| Protocol | Source Port | Source Address | Destination Port | Destination Address |
|---|---|---|---|---|
| UDP | 1024-65535 | Service Link IP | 53 | DHCP provided DNS server |
| UDP | 443, 1024-65535 | Service Link IP | 443 | AWS Outposts Service Link endpoints |
| TCP | 1024-65535 | Service Link IP | 443 | AWS Outposts Registration endpoints |

**Note**

Instances in an Outpost cannot use the service link to communicate with instances in another Outposts. Leverage routing through the local gateway or local network interface to communicate between Outposts.

## Updates and the service link

AWS maintains a secure network connection between your Outpost server and its parent AWS Region. This network connection, called the service link, is essential in managing the Outpost by providing intra-VPC traffic between the Outpost and AWS Region. AWS Well-Architected best practices recommend deploying applications across two Outposts parented to different Availability Zones with an active-active design. For more information, see AWS Outposts High Availability Design and Architecture Considerations.

The service link is regularly updated to maintain operational quality and performance. During maintenance, you might observe brief periods of latency and packet loss on this network resulting in impact on workloads that are dependent on VPC connectivity to resources hosted in-region. However, traffic traversing the Local Network Interfaces (LNI) will not be impacted. You can avoid

impact to your application by following AWS Well-Architected best practices and by ensuring your applications are resilient to failures or maintenance activities affecting a single Outpost server.

## Redundant internet connections

When you build connectivity from your Outpost to the AWS Region, we recommend that you create multiple connections for higher availability and resiliency. For more information, see AWS Direct Connect Resiliency Recommendations.

If you need connectivity to the public internet, you can use redundant internet connections and diverse internet providers, just as you would with your existing on-premises workloads.

# Outposts and sites

Manage Outposts and sites for AWS Outposts.

You can tag Outposts and sites to help you identify them or categorize them according to your organization's needs. For more information about tagging, see Tagging AWS Resources in the *AWS General Reference Guide*.

**Topics**



# Manage Outposts

AWS Outposts includes hardware and virtual resources known as Outposts. Use this section to create and manage Outposts, including changing the name, and adding or viewing details or tags.

**To create an Outpost**

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Outposts**.
4. Choose **Create Outpost**.
5. Choose a hardware type for this Outpost.
6. Enter a name and description for your Outpost.
7. Select an Availability Zone for your Outpost.
8. (Optional) Choose **Private connectivity option**. For **VPC** and **Subnet**, select a VPC and subnet in the same AWS account and Availability Zone as your Outpost.

   > **Note**
   >
   > If you need to undo the private connectivity for your Outpost, you must contact AWS Enterprise Support.

9. From **Site ID**, do one of the following:

- To select an existing site, choose the site.
- To create a new site, choose **Create site**, click **Next**, and enter the information about your site in the new window.

  After you create the site, return to this window to select the site. You may need to refresh the site list to see the new site. To refresh your data, choose the refresh icon (  ).

  For more information, see the section called “Sites”.

10. Choose **Create Outpost**.

> **Tip**
>
> To add capacity to your new Outpost, you must place an order.

Use the following steps to edit the name and description of an Outpost.

**To edit the Outpost name and description**

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Outposts**.
4. Select the Outpost, and then choose **Actions**, **Edit Outpost**.
5. Modify the name and description.

   For **Name**, enter the name.

   For **Description**, enter the description.
6. Choose **Save changes**.

Use the following steps to view the details of an Outpost.

**To view the Outpost details**

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.

2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Outposts**.
4. Select the Outpost, and then choose **Actions, View details**.

You can also use the AWS CLI to view Outpost details.

**To view Outpost details with the AWS CLI**

- Use the get-outpost AWS CLI command.

Use the following steps to manage tags on an Outpost.

**To manage the Outpost tags**

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Outposts**.
4. Select the Outpost, and then choose **Actions**, **Manage tags**.
5. Add or remove a tag.

   To add a tag, choose **Add new tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   To remove a tag, choose **Remove** to the right of the tag's key and value.
6. Choose **Save changes**.

# Manage Outpost sites

The customer-managed physical buildings where AWS will install your Outpost. A site must meet the facility, networking, and power requirements for your Outpost. For more information, see *Requirements*.

**To create an Outpost site**

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Sites**.
4. Choose **Create site**.
5. Choose a supported hardware type for the site.
6. Enter a name, description, and operating address for your site. If you chose to support racks at the site, enter the following information:
   - **Max weight** – Specify the maximum rack weight that this site can support.
   - **Power draw** – Specify in kVA the power draw available at the hardware placement position for the rack.
   - **Power option** – Specify the power option that you can provide for hardware.
   - **Power connector** – Specify the power connector that AWS should plan to provide for connections to the hardware.
   - **Power feed drop** – Specify whether the power feed comes above or below the rack.
   - **Uplink speed** – Specify the uplink speed the rack should support for the connection to the Region.
   - **Number of uplinks** – Specify the number of uplinks for each Outpost network device that you intend to use to connect the rack to your network.
   - **Fiber type** – Specify the type of fiber that you will use to attach the Outpost to your network.
   - **Optical standard** – Specify the type of optical standard that you will use to attach the Outpost to your network.
   - **Notes** – Specify notes about a site.
7. Read the facility requirements and choose **I have read the facility requirements**.
8. Choose **Create site**.

Use the following steps to edit an Outpost site.

**To edit a site**

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.

2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Sites**.
4. Select the site, and then select **Actions**, **Edit site**.
5. You can modify the name, description, operating address, and site details.

   If you change the operating address, be aware that the changes will not propagate to existing orders.
6. Choose **Save changes**.

Use the following steps to view the details of an Outpost site.

**To view the site details**

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Sites**.
4. Select the site, and then choose **Actions**, **View details**.

Use the following steps to manage tags on an Outpost site.

**To manage the site tags**

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Sites**.
4. Select the site, and then choose **Actions**, **Manage tags**.
5. Add or remove a tag.

   To add a tag, choose **Add new tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   To remove a tag, choose **Remove** to the right of the tag's key and value.
6. Choose **Save changes**.

# Return an AWS Outposts server

If AWS Outposts detects a defect in the server, we will inform you, start the replacement process to send you a new server, and provide you with the shipping label through the AWS Outposts console.

If you want to return the server because the server reaches the end of the contract term or for any other reason, contact AWS Support Center.

**Topics**



The following steps explain how to return a server to AWS.

## 1. Prepare the server for return

To prepare the server for return, unshare resources, backup data, delete local network interfaces and terminate active instances.

1. If the Outpost's resources are shared, you must unshare these resources.

   You can unshare a shared Outpost resource in one of the following ways:

   - Use the AWS RAM console. For more information, see Updating a resource share in the *AWS RAM User Guide*.
   - Use the AWS CLI to run the disassociate-resource-share command.

   For the list of Outpost resources that can be shared, see Shareable Outpost resources.
2. Create backups of the data stored in the instance storage of the Amazon EC2 instances running on the AWS Outposts server.
3. Delete the local network interfaces associated with the instances that were running on the server.

4. Terminate the active instances associated with subnets on your Outpost. To terminate the instances, follow the instructions in Terminate your instance in the *Amazon EC2 User Guide for Linux Instances*.

## 2. Obtain the return shipping label

> **Important**
>
> You must only use the shipping label that AWS provides. Do not create your own shipping label.

Obtain your shipping label based on the reason for your return.

Shipping label for a server that is being replaced

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Orders**.
3. Under **Replacement order summary**, choose **Print return label** and choose the configuration ID of the server that you plan to return.

Shipping label for a server that is not being replaced

1. Contact AWS Support Center.
2. Request a shipping label for the server you intend to return.

## 3. Pack the server

To pack your server, use the box and packaging material that the server originally came in. You can also use the box the replacement server arrives in. Alternatively, contact AWS Support Center to request a box. After packing the server, affix the shipping label that AWS provided.

# 4. Return the server through the courier

You must return the server through the designated courier for your country. You can deliver the server to the courier or schedule the day and time that you prefer for the courier to pick up the server. The shipping label that AWS provides contains the correct address to return the server.

The following table shows who to contact for the country you are shipping from:

| Country | Contact |
| --- | --- |
| Argentina | Contact AWS Support Center. In your request, include the following information:<br>• The tracking number that is on the AWS-provided shipping label<br>• The date and time that you prefer the courier to pick up the server<br>• A contact name<br>• A phone number<br>• An email address |
| Bahrain | |
| Brazil | |
| Brunei | |
| Canada | |
| Chile | |
| Colombia | |
| Hong Kong | |
| India | |
| Indonesia | |
| Japan | |
| Malaysia | |
| Nigeria | |
| Oman | |
| Panama | |
| Peru | |

| Country | Contact |
| --- | --- |
| Philippines | |
| Serbia | |
| Singapore | |
| South Africa | |
| South Korea | |
| Taiwan | |
| Thailand | |
| United Arab Emirates | |
| Vietnam | |
| United States of America | Contact UPS.<br><br>You can return the server in the following ways:<br><br>• Return the server during a routine UPS pickup at your site.<br>• Drop-off the server at a UPS location.<br>• Schedule a pickup for a date and time you prefer. Enter the tracking number from the AWS-provided shipping label for free shipping. |

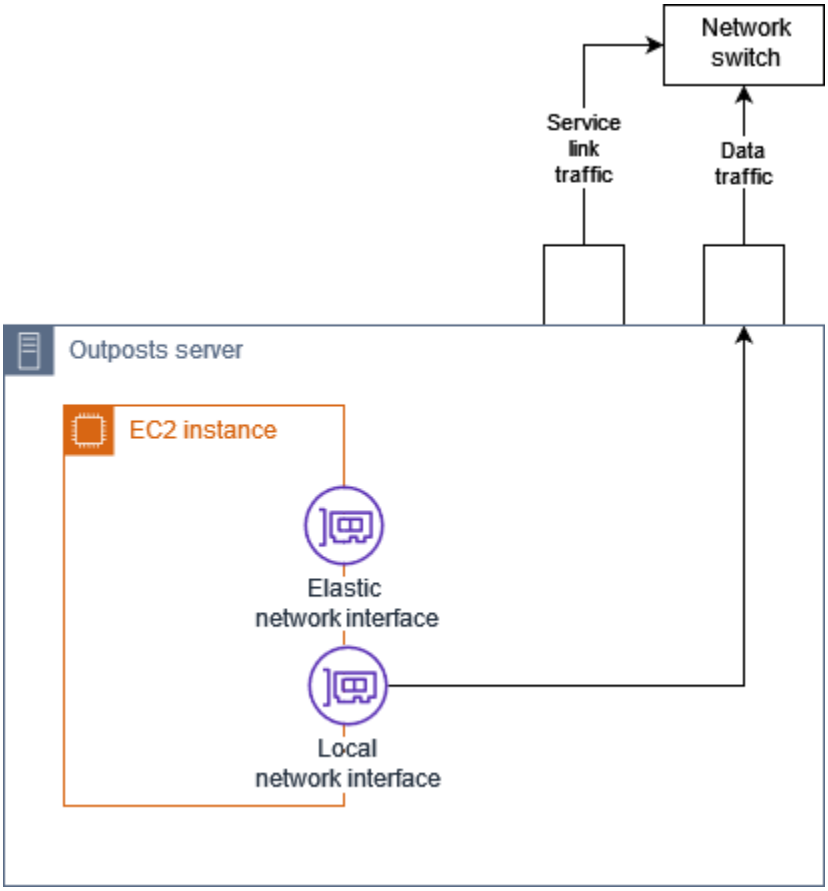
| Country | Contact |
| --- | --- |
| All other countries | Contact DHL.<br><br>You can return the server in the following ways:<br><br>• Drop-off the server at a DHL location.<br>• Schedule a pickup for a date and time you prefer. Enter the DHL Waybill number from the AWS-provided shipping label for free shipping.<br><br>If you get the following error `Courier pickup cannot be scheduled for an import shipment`, it usually means that the pickup country that you selected does not match the pickup country on the return shipment label. Select the country where the shipment originates from and try again. |

# Local network interfaces

With AWS Outposts servers, a *local network interface* (LNI) is a logical networking component that connects the Amazon EC2 instances in your Outposts subnet to your on-premises network.

A local network interface runs directly on your local area network. With this type of local connectivity, you don't need routers or gateways to communicate with your on-premises equipment. Local network interfaces are named similarly to network interfaces or elastic network interfaces. We distinguish between the two interfaces by always using *local* when we refer to local network interfaces.

After you enable local network interfaces on an Outpost subnet, you can configure the EC2 instances in the Outpost subnet to include a local network interface in addition to the elastic network interface. The local network interface connects to the on-premises network while the network interface connects to the VPC. The following diagram shows an EC2 instance on an Outposts server with both an elastic network interface and a local network interface.



You must configure the operating system to enable the local network interface to communicate on your local area network, just as you would for any other on-premises equipment. You can't use DHCP option sets in a VPC to configure a local network interface because a local network interface runs on your local area network.

The elastic network interface works exactly as it does for instances in an Availability Zone subnet. For example, you can use the VPC network connection to access the public Regional endpoints for AWS services, or you can use interface VPC endpoints to access AWS services using AWS PrivateLink. For more information, see AWS Outposts connectivity to AWS Regions.

**Contents**



# Local network interface basics

Local network interfaces provide access to a physical layer-two network. A VPC is a virtualized layer-three network. Local network interfaces do not support VPC networking components. These components include security groups, network access control lists, virtualized routers or route tables, and flow logs. The local network interface does not provide the Outpost server with visibility into VPC layer-three flows. The host operating system of the instance does have full visibility into frames from the physical network. You can apply standard firewall logic to information within these frames. However, this communication happens inside the instance but outside the purview of the virtualized constructs.

**Considerations**

- Local network interfaces support ARP and DHCP protocols. They do not support general L2 broadcast messages.
- Quotas for local network interfaces comes out of your quota for network interfaces. For more information, see Network interfaces in the *Amazon VPC User Guide*.
- Each EC2 instance can have one local network interface.
- A local network interface can't use the primary network interface (eth0) of the instance.
- Outposts servers can host multiple EC2 instances, each with a local network interface.

> **Note**
>
> EC2 instances within the same server can communicate directly without sending data outside the Outposts server. This communication includes traffic over a local network interface or elastic network interfaces.

- Local network interfaces are available only for instances running in an Outposts subnet on an Outpost server.
- Local network interfaces do not support promiscuous mode or MAC address spoofing.

## Performance

The LNI of each instance size provides a portion of the physical 10 GbE LNI available bandwidth. The following table lists the LNI network performance for each instance type:

| Instance type | Baseline bandwidth (Gbps) | Burst bandwidth (Gbps) |
|---|---|---|
| c6id.large | 0.15625 | 2.5 |
| c6id.large | 0.15625 | 2.5 |
| c6id.xlarge | 0.3125 | 2.5 |
| c6id.2xlarge | 0.625 | 2.5 |
| c6id.4xlarge | 1.25 | 2.5 |
| c6id.8xlarge | 2.5 | 2.5 |
| c6id.12xlarge | 3.75 | 3.75 |
| c6id.16xlarge | 5 | 5 |
| c6id.24xlarge | 7.5 | 7.5 |
| c6id.32xlarge | 10 | 10 |
| c6gd.medium | 0.15625 | 4 |

| Instance type | Baseline bandwidth (Gbps) | Burst bandwidth (Gbps) |
|---|---|---|
| c6gd.large | 0.3125 | 4 |
| c6gd.xlarge | 0.625 | 4 |
| c6gd.2xlarge | 1.25 | 4 |
| c6gd.4xlarge | 2.5 | 4 |
| c6gd.8xlarge | 4.8 | 4.8 |
| c6gd.12xlarge | 7.5 | 7.5 |
| c6gd.16xlarge | 10 | 10 |

## Security groups

By design, the local network interface does not use security groups in your VPC. A security group controls inbound and outbound *VPC traffic*. The local network interface is not attached to the VPC. The local network interface is attached to your local network. To control inbound and outbound traffic on the local network interface, use a firewall or similar strategy, just as you would with the rest of your on-premises equipment.

## Monitoring

CloudWatch metrics are produced for each local network interface, just as they are for elastic network interfaces. For more information for Linux instances, see Monitor network performance for your EC2 instance in the *Amazon EC2 User Guide for Linux Instances*. For Windows instances, see Monitor network performance for your EC2 instance in the *Amazon EC2 User Guide for Windows Instances*.

## MAC addresses

AWS provides MAC addresses for local network interfaces. Local network interfaces use locally administered addresses (LAA) for their MAC addresses. A local network interface uses the same MAC address until you delete the interface. After you delete a local network interface, remove the MAC address from your local configurations. AWS can reuse MAC addresses that are no longer in use.

# Enable subnets on Outposts servers for local network interfaces

Use the modify-subnet-attribute command from the AWS CLI to enable an Outpost subnet for local network interfaces. You must specify the position of the network interface on the device index. All instances launched in an enabled Outpost subnet use this device position for local network interfaces. For example, a value of 1 indicates that the secondary network interface (eth1) for an instance in the Outpost subnet is the local network interface.

**To enable an Outpost subnet for local network interfaces**

At a command prompt, use the following command to specify the device position for the local network interface.

```
aws ec2 modify-subnet-attribute \
    --subnet-id subnet-1a2b3c4d \
    --enable-lni-at-device-index 1
```

# Work with local network interfaces

Use this section to understand how to work with local network interfaces.

**Tasks**

- Add a local network interface
- View the local network interface
- Configure the operating system

## Add a local network interface

You can add a local network interface (LNI) to an Amazon EC2 instance on an Outposts subnet during or after launch. You do so by adding a secondary network interface to the instance, using the device index that you specified when you enabled the Outpost subnet for local network interfaces.

**Consideration**

When you specify the secondary network interface using the console, the network interface is created using device index 1. If this is not the device index that you specified when you enabled

the Outpost subnet for local network interfaces, you can specify the correct device index by using the AWS CLI or an AWS SDK instead. For example, use the following commands from the AWS CLI: create-network-interface and attach-network-interface.

**To add an LNI during instance launch**

1. In the launch instance wizard, choose **Edit** next to **Network settings**.
2. Expand **Advanced network configuration**.
3. Choose **Add network interface**. This creates a network interface using device index 1. If you specified 1 as the LNI device index for the Outpost subnet, then this network interface will be the local network interface for the instance.
4. Choose the Outpost subnet, and update the configuration for the network interface as needed.
5. Complete the wizard to launch the instance.

**To add an LNI after instance launch**

1. In the navigation pane, choose **Network and Security**, **Network Interfaces**.
2. **Create the network interface**
   a. Choose **Create network interface**.
   b. Select the same Outpost subnet as the instance.
   c. Verify that **Private IPv4 address** is set to **Auto-assign**.
   d. Select any security group. Security groups do not apply to LNIs, so the security group that you select is not relevant.
   e. Choose **Create network interface**.
3. **Attach the network interface to the instance**
   a. Select the check box for the newly created network interface.
   b. Choose **Actions**, **Attach**.
   c. Choose the instance.
   d. Choose **Attach**. The network interface is attached at device index 1. If you specified 1 as the LNI device index for the Outpost subnet, then this network interface is the local network interface for the instance.

## View the local network interface

While the instance is in the running state, you can use the Amazon EC2 console to view both the elastic network interface and the local network interface for the instances in your Outpost subnet. Select the instance and choose the **Networking** tab.

The console displays a private IPv4 address for the LNI from the subnet CIDR. This address is not the IP address of the LNI, and it is not usable. However, this address is allocated from the subnet CIDR, so you must account for it in your subnet sizing. You must set the IP address for the LNI within the guest operating system, either statically or through your DHCP server.

## Configure the operating system

After you enable local network interfaces, Amazon EC2 instances will have two network interfaces, one of which is a local network interface. Ensure that you configure the operating system of the Amazon EC2 instances that you launch to support a multi-homed networking configuration.

# Local network connectivity for servers

Use this topic to understand the network cabling and topology requirements for hosting an Outpost server. For more information, see Local network interfaces.

**Contents**



## Server topology on your network

An Outpost server requires two distinct connections to your networking equipment. Each connection uses a different cable and carries a different type of traffic. The multiple cables are for traffic-class isolation only, and not for redundancy. The two cables do not need to connect to a common network.

The following table describes Outpost server traffic types and labels.

| Traffic label | Description |
| --- | --- |
| **2** | **Service link traffic** – This traffic enables communication between the Outpost and the AWS Region for both management of the Outpost and intra-VPC traffic between the AWS Region and the Outpost. Service link traffic includes the service link connectio n from the Outpost to the Region. The service link is a custom VPN or VPNs from the Outpost to the Region. The Outpost connects to the Availability Zone in the Region that you chose at time of purchase. |
| **1** | **Local network interface (LNI) link traffic** – This traffic enables communication from your VPC to your local LAN over the local network interface. Local link traffic includes instances running on the Outpost that communicate with your on-premises network. Local link traffic can also include instances communica ting with the internet through your on-premis es network. |

## Server physical connectivity

Each Outpost server includes non-redundant physical uplink ports. Ports have their own speed and connector requirements as follows:

- **10Gbe** – connector type QSFP+

### QSFP+ cable

The QSFP+ cable has a connector that you attach to port 3 on the Outpost server. The other end of the QSFP+ cable has four SFP+ interfaces that you connect to your switch. Two of the switch-side

interfaces are labeled 1 and 2. Both the interfaces are required for an Outpost server to function. Use the 2 interface for service link traffic and the 1 interface for LNI link traffic. The remaining interfaces are not used.

## Service link traffic for servers

Configure the service link port on your switch as an untagged access port to a VLAN with a gateway and a route to the following Region endpoints:

- Service link endpoints
- Outposts registration endpoint

The service link connection must have public DNS available for the Outpost to discover its registration endpoint in the AWS Region. The connection can have a NAT device between the Outpost server and the registration endpoint. For more information about the public address ranges for AWS, see AWS IP address ranges in the *Amazon VPC User Guide* and AWS Outposts endpoints and quotas in the *AWS General Reference*.

To register the server, open the following network ports:

- TCP 443
- UDP 443
- UDP 53

### Uplink speed

Each Outposts server requires a minimum uplink speed of 20 Mbps to the AWS Region.

You may need a faster uplink depending on your LNI link and service link utilization. For more information, see Bandwidth recommendations for service links.

## Local network interface (LNI) link traffic

Configure the LNI link port on your upstream network device as a standard access port to a VLAN on your local network. If you have more than one VLAN, configure all the ports on the upstream network device as trunk ports. Configure the port on your upstream network device to expect multiple MAC addresses. Each instance launched on the server will use a MAC address. Some

network devices offer port-security features that will shut down a port that reports multiple MAC addresses.

> **Note**
>
> AWS Outposts servers do not tag VLAN traffic. If you configure your LNI as trunk, you must ensure that your OS tags VLAN traffic.

The following example shows how to configure VLAN tagging for your LNI on Amazon Linux 2023. If you are using another Linux distribution, see the documentation for your Linux distribution about configuring VLAN tagging.

**Example: To configure VLAN tagging for your LNI on Amazon Linux 2023 and Amazon Linux 2**

1. Ensure that the 8021q module is loaded into the kernel. If not, load it using the `modprobe` command.

```
modinfo 8021q
modprobe --first-time 8021q
```

2. Create the VLAN device. In this example:

   - The interface name of the LNI is `ens6`
   - The VLAN id is 59
   - The name assigned for the VLAN device is `ens6.59`

```
ip link add link ens6 name ens6.59 type vlan id 59
```

3. Optional. Complete this step if you want to manually assign the IP. In this example we are assigning the IP 192.168.59.205, where the subnet CIDR is 192.168.59.0/24.

```
ip addr add 192.168.59.205/24 brd 192.168.59.255 dev ens6.59
```

4. Activate the link.

```
ip link set dev ens6.59 up
```

To configure your network interfaces at the OS level and make the VLAN tagging changes persistent, refer to the following resources:

- If you are using Amazon Linux 2, see Configure your network interface using ec2-net-utils for Amazon Linux in the *Amazon EC2 User Guide for Linux Instances*.
- If you are using Amazon Linux 2023, see Networking service in the *Amazon Linux 2023 User Guide*.

## Server IP address assignment

You do not need public IP address assignments for Outpost servers.

Dynamic host control protocol (DHCP) is a network management protocol used to automate the process of configuring devices on IP networks. In the context of Outpost servers, you can use DHCP two ways:

- Network cards on the server
- Local network interfaces on instances

For service link, Outpost servers use DHCP to attach to the local network. DHCP must return DNS name servers and a default gateway. Outpost servers do not support static IP assignment of service link.

For LNI link, use DHCP to configure instances to be attached to your local network. For more information see, the section called “Configure the operating system”.

> **Note**
>
> Ensure that you use a stable IP address for the Outpost server. IP address changes can cause temporary service disruptions on the Outpost subnet.

## Server registration

When Outpost servers establish a connection on the local network, they use the service link connection to connect to Outpost registration endpoints and register themselves. Registration requires public DNS. When servers register, they create a secure tunnel to their service link

endpoint in the Region. Outpost servers use TCP port 443 to facilitate communication with the Region over the public internet. Currently, AWS Outposts servers do not support private connectivity through VPC. For more information, see the section called “Step 6: Authorize server”.

# Working with shared AWS Outposts resources

With Outpost sharing, Outpost owners can share their Outposts and Outpost resources, including Outpost sites and subnets, with other AWS accounts under the same AWS organization. As an Outpost owner, you can create and manage Outpost resources centrally, and share the resources across multiple AWS accounts within your AWS organization. This allows other consumers to use Outpost sites, configure VPCs, and launch and run instances on the shared Outpost.

In this model, the AWS account that owns the Outpost resources (*owner*) shares the resources with other AWS accounts (*consumers*) in the same organization. Consumers can create resources on Outposts that are shared with them in the same way that they would create resources on Outposts that they create in their own account. The owner is responsible for managing the Outpost and resources that they create in it. Owners can change or revoke shared access at any time. With the exception of instances that consume Capacity Reservations, owners can also view, modify, and delete resources that consumers create on shared Outposts. Owners cannot modify instances that consumers launch into Capacity Reservations that they have shared.

Consumers are responsible for managing the resources that they create on Outposts that are shared with them, including any resources that consume Capacity Reservations. Consumers can't view or modify resources owned by other consumers or by the Outpost owner. They also can't modify Outposts that are shared with them.

An Outpost owner can share Outpost resources with:

- Specific AWS accounts inside of its organization in AWS Organizations.
- An organizational unit inside of its organization in AWS Organizations.
- Its entire organization in AWS Organizations.

**Contents**

- Identifying a shared Outpost resource
- Shared Outpost resource permissions
- Billing and metering
- Limitations

## Shareable Outpost resources

An Outpost owner can share the Outpost resources listed in this section with consumers.

These are the resources available for Outpost servers. For rack resources, see Working with shared AWS Outposts resources in the AWS Outposts User Guide for Outposts rack.

- **Allocated Dedicated Hosts** – Consumers with access to this resource can:
  - Launch and run EC2 instances on a Dedicated Host.
- **Outposts** – Consumers with access to this resource can:
  - Create and manage subnets on the Outpost.
  - Use the AWS Outposts API to view information about the Outpost.
- **Sites** – Consumers with access to this resource can:
  - Create, manage, and control an Outpost at the site.
- **Subnets** – Consumers with access to this resource can:
  - View information about subnets.
  - Launch and run EC2 instances in subnets.

  Use the Amazon VPC console to share an Outpost subnet. For more information, see Sharing a subnet in the *Amazon VPC User Guide*.

## Prerequisites for sharing Outposts resources

- To share an Outpost resource with your organization or an organizational unit in AWS Organizations, you must enable sharing with AWS Organizations. For more information, see Enable Sharing with AWS Organizations in the *AWS RAM User Guide*.
- To share an Outpost resource, you must own it in your AWS account. You cannot share an Outpost resource that has been shared with you.
- To share an Outpost resource, you must share it with an account that is within your organization.

# Related services

Outpost resource sharing integrates with AWS Resource Access Manager (AWS RAM). AWS RAM is a service that enables you to share your AWS resources with any AWS account or through AWS Organizations. With AWS RAM, you share resources that you own by creating a *resource share*. A resource share specifies the resources to share, and the consumers with whom to share them. Consumers can be individual AWS accounts, organizational units, or an entire organization in AWS Organizations.

For more information about AWS RAM, see the *AWS RAM User Guide*.

# Sharing across Availability Zones

To ensure that resources are distributed across the Availability Zones for a Region, we independently map Availability Zones to names for each account. This could lead to Availability Zone naming differences across accounts. For example, the Availability Zone `us-east-1a` for your AWS account might not have the same location as `us-east-1a` for another AWS account.

To identify the location of your Outpost resource relative to your accounts, you must use the *Availability Zone ID* (AZ ID). The AZ ID is a unique and consistent identifier for an Availability Zone across all AWS accounts. For example, `use1-az1` is an AZ ID for the `us-east-1` Region and it is the same location in every AWS account.

**To view the AZ IDs for the Availability Zones in your account**

1. Open the AWS RAM console at https://console.aws.amazon.com/ram.
2. The AZ IDs for the current Region are displayed in the **Your AZ ID** panel on the right-hand side of the screen.

> **Note**
>
> Local gateway route tables are in the same AZ as their Outpost, so you do not need to specify an AZ ID for route tables.

# Sharing an Outpost resource

When an owner shares an Outpost with a consumer, the consumer can create resources on the Outpost in the same way that they would create resources on Outposts that they create in their own account. Consumers with access to shared local gateway route tables can create and manage VPC associations. For more information, see Shareable Outpost resources.

To share an Outpost resource, you must add it to a resource share. A resource share is an AWS RAM resource that lets you share your resources across AWS accounts. A resource share specifies the resources to share, and the consumers with whom they are shared. When you share an Outpost resource using the AWS Outposts console, you add it to an existing resource share. To add the Outpost resource to a new resource share, you must first create the resource share using the AWS RAM console.

If you are part of an organization in AWS Organizations and sharing within your organization is enabled, you can grant consumers in your organization access from the AWS RAM console to the shared Outpost resource. Otherwise, consumers receive an invitation to join the resource share and are granted access to the shared Outpost resource after accepting the invitation.

You can share an Outpost resource that you own using the AWS Outposts console, AWS RAM console, or the AWS CLI.

**To share an Outpost that you own using the AWS Outposts console**

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions**, **View details**.
4. On the **Outpost summary** page, choose **Resource shares**.
5. Choose **Create resource share**.

You are redirected to the AWS RAM console to finish sharing the Outpost using the following procedure. To share a local gateway route table that you own, use the following procedure as well.

**To share an Outpost or local gateway route table that you own using the AWS RAM console**

See Creating a Resource Share in the *AWS RAM User Guide*.

**To share an Outpost or local gateway route table that you own using the AWS CLI**

Use the create-resource-share command.

# Unsharing a shared Outpost resource

When a shared Outpost is unshared, consumers can no longer view the Outpost in the AWS Outposts console. They cannot create new subnets on the Outpost, create new EBS volumes on the Outpost, or view the Outpost details and instance types using the AWS Outposts console or the AWS CLI. Existing subnets, volumes, or instances created by consumers are not deleted. Any existing subnets consumers created on the Outpost can still be used to launch new instances.

When a shared local gateway route table is unshared, consumers can no longer create new VPC associations to it. Any existing VPC associations consumers created remain associated with the route table. Resources in these VPCs can continue to route traffic to the local gateway.

To unshare a shared Outpost resource that you own, you must remove it from the resource share. You can do this using the AWS RAM console or the AWS CLI.

**To unshare a shared Outpost resource that you own using the AWS RAM console**

See Updating a Resource Share in the *AWS RAM User Guide*.

**To unshare a shared Outpost resource that you own using the AWS CLI**

Use the disassociate-resource-share command.

# Identifying a shared Outpost resource

Owners and consumers can identify shared Outposts using the AWS Outposts console and AWS CLI. They can identify shared local gateway route tables using the AWS CLI.

**To identify a shared Outpost using the AWS Outposts console**

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions**, **View details**.
4. On the **Outpost summary** page, view the **Owner ID** to identify the AWS account ID of the Outpost owner.

**To identify a shared Outpost resource using the AWS CLI**

Use the list-outposts and describe-local-gateway-route-tables commands. These commands return the Outpost resources that you own and Outpost resources that are shared with you. `OwnerId` shows the AWS account ID of the Outpost resource owner.

# Shared Outpost resource permissions

## Permissions for owners

Owners are responsible for managing the Outpost and resources that they create in it. Owners can change or revoke shared access at any time. They can use AWS Organizations to view, modify, and delete resources that consumers create on shared Outposts.

## Permissions for consumers

Consumers can create resources on Outposts that are shared with them in the same way that they would create resources on Outposts that they create in their own account. Consumers are responsible for managing the resources that they launch onto Outposts that are shared with them. Consumers can't view or modify resources owned by other consumers or by the Outpost owner, and they can't modify Outposts that are shared with them.

# Billing and metering

Owners are billed for Outposts and Outpost resources that they share. They are also billed for any data transfer charges associated with their Outpost's service link VPN traffic from the AWS Region.

There are no additional charges for sharing local gateway route tables. For shared subnets, the VPC owner is billed for VPC-level resources such as AWS Direct Connect and VPN connections, NAT gateways, and Private Link connections.

Consumers are billed for application resources that they create on shared Outposts, such as load balancers and Amazon RDS databases. Consumers are also billed for chargeable data transfers from the AWS Region.

# Limitations

The following limitations apply to working with AWS Outposts sharing:

- Limitations for shared subnets apply to working with AWS Outposts sharing. For more information about VPC sharing limits, see Limitations in the *Amazon Virtual Private Cloud User Guide*.
- Service quotas apply per individual account.

# Security in AWS Outposts

Security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to AWS Outposts, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

For more information about security and compliance for AWS Outposts, see the AWS Outposts servers FAQ.

This documentation helps you understand how to apply the shared responsibility model when using AWS Outposts. It shows you how to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your resources.

**Contents**



# Data protection in AWS Outposts

The AWS shared responsibility model applies to data protection in AWS Outposts. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the

AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties.

For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

### Encryption at rest

With AWS Outposts, all data is encrypted at rest. The key material is wrapped to an external key stored in a removable device, the Nitro Security Key (NSK). The NSK is required to decrypt the data on your Outpost servers.

### Encryption in transit

AWS encrypts in-transit data between your Outpost and its AWS Region. For more information, see Connectivity through service links.

### Data deletion

When you terminate an EC2 instance, the memory allocated to it is scrubbed (set to zero) by the hypervisor before it is allocated to a new instance, and every block of storage is reset.

Destroying the Nitro Security Key cryptographically shreds the data on your Outpost. For more information, see Cryptographically shred server data.

## Identity and access management (IAM) for AWS Outposts

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be authenticated (signed in) and authorized (have permissions) to use AWS Outposts resources. You can use IAM for no additional charge.

**Contents**

- AWS Outposts policy examples
- Using service-linked roles for AWS Outposts
- AWS managed policies for AWS Outposts

## How AWS Outposts works with IAM

Before you use IAM to manage access to AWS Outposts, learn what IAM features are available to use with AWS Outposts.

**IAM features you can use with AWS Outposts**

| IAM feature | AWS Outposts support |
|---|---|
| Identity-based policies | Yes |
| Resource-based policies | No |
| Policy actions | Yes |
| Policy resources | Yes |
| Policy condition keys (service-specific) | Yes |
| ACLs | No |
| ABAC (tags in policies) | Yes |
| Temporary credentials | Yes |
| Principal permissions | Yes |
| Service roles | No |
| Service-linked roles | Yes |

## Identity-based policies for AWS Outposts

| | |
|---|---|
| Supports identity-based policies | Yes |

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

**Identity-based policy examples for AWS Outposts**

To view examples of AWS Outposts identity-based policies, see AWS Outposts policy examples.

## Resource-based policies within AWS Outposts

| | |
|---|---|
| Supports resource-based policies | No |

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

## Policy actions for AWS Outposts

| | |
|---|---|
| Supports policy actions | Yes |

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of AWS Outposts actions, see Actions defined by AWS Outposts in the *Service Authorization Reference*.

Policy actions in AWS Outposts use the following prefix before the action:

```
outposts
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
    "outposts:action1",
    "outposts:action2"
]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `List`, include the following action:

```
"Action": "outposts:List*"
```

## Policy resources for AWS Outposts

| | |
|---|---|
| Supports policy resources | Yes |

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

Some AWS Outposts API actions support multiple resources. To specify multiple resources in a single statement, separate the ARNs with commas.

```
"Resource": [
    "resource1",
    "resource2"
]
```

To see a list of AWS Outposts resource types and their ARNs, see Resource types defined by AWS Outposts in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by AWS Outposts.

## Policy condition keys for AWS Outposts

| | |
|---|---|
| Supports service-specific policy condition keys | Yes |

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple

values for a single condition key, AWS evaluates the condition using a logical OR operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of AWS Outposts condition keys, see Condition keys for AWS Outposts in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by AWS Outposts.

To view examples of AWS Outposts identity-based policies, see AWS Outposts policy examples.

## ACLs in AWS Outposts

| | |
|---|---|
| Supports ACLs | No |

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

## ABAC with AWS Outposts

| | |
|---|---|
| Supports ABAC (tags in policies) | Yes |

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see What is ABAC? in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

## Using temporary credentials with AWS Outposts

| | |
|---|---|
| Supports temporary credentials | Yes |

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switching to a role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

## Cross-service principal permissions for AWS Outposts

| | |
|---|---|
| Supports forward access sessions (FAS) | Yes |

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a

different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

### Service roles for AWS Outposts

| | |
|---|---|
| Supports service roles | No |

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

### Service-linked roles for AWS Outposts

| | |
|---|---|
| Supports service-linked roles | Yes |

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing AWS Outposts service-linked roles, see Using service-linked roles for AWS Outposts.

## AWS Outposts policy examples

By default, users and roles don't have permission to create or modify AWS Outposts resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Creating IAM policies in the *IAM User Guide*.

For details about actions and resource types defined by AWS Outposts, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS Outposts in the *Service Authorization Reference*.

**Contents**



## Policy best practices

Identity-based policies determine whether someone can create, access, or delete AWS Outposts resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see IAM Access Analyzer policy validation in the *IAM User Guide*.

- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Configuring MFA-protected API access in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

## Example: Using resource-level permissions

The following example uses resource-level permissions to grant permission to get information about the specified Outpost.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "outposts:GetOutpost",
      "Resource": "arn:aws:outposts:region:12345678012:outpost/op-1234567890abcdef0"
    }
  ]
}
```

The following example uses resource-level permissions to grant permission to get information about the specified site.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "outposts:GetSite",
      "Resource": "arn:aws:outposts:region:12345678012:site/os-0abcdef1234567890"
    }
  ]
}
```

# Using service-linked roles for AWS Outposts

AWS Outposts uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to AWS Outposts. Service-linked roles are predefined by AWS Outposts and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up your AWS Outposts more efficient because you don't have to manually add the necessary permissions. AWS Outposts defines the permissions of its service-linked roles, and unless defined otherwise, only AWS Outposts can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting the related resources. This protects your AWS Outposts resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

## Service-linked role permissions for AWS Outposts

AWS Outposts uses the service-linked role named **AWSServiceRoleForOutposts_*OutpostID*** – Allows Outposts to access AWS resources for private connectivity on your behalf. This service-linked role allows private connectivity configuration, creates network interfaces, and attaches them to service link endpoint instances.

The AWSServiceRoleForOutposts_*OutpostID* service-linked role trusts the following services to assume the role:

- `outposts.amazonaws.com`

The **AWSServiceRoleForOutposts_*OutpostID*** service-linked role includes the following policies:

- **AWSOutpostsServiceRolePolicy**
- **AWSOutpostsPrivateConnectivityPolicy_*OutpostID***

The **AWSOutpostsServiceRolePolicy** policy is a service-linked role policy to enable access to AWS resources managed by AWS Outposts.

This policy allows AWS Outposts to complete the following actions on the specified resources:

- Action: `ec2:DescribeNetworkInterfaces` on `all AWS resources`
- Action: `ec2:DescribeSecurityGroups` on `all AWS resources`
- Action: `ec2:CreateSecurityGroup` on `all AWS resources`
- Action: `ec2:CreateNetworkInterface` on `all AWS resources`

The **AWSOutpostsPrivateConnectivityPolicy_*OutpostID*** policy allows AWS Outposts to complete the following actions on the specified resources:

- Action: `ec2:AuthorizeSecurityGroupIngress` on `all AWS resources that match the following Condition:`

```
{ "StringLike" : { "ec2:ResourceTag/outposts:private-connectivity-resourceId" :
 "OutpostID" }} and { "StringEquals" : { "ec2:Vpc" : "vpcArn" }}
```

- Action: `ec2:AuthorizeSecurityGroupEgress` on `all AWS resources that match the following Condition:`

```
{ "StringLike" : { "ec2:ResourceTag/outposts:private-connectivity-resourceId" :
 "OutpostID" }} and { "StringEquals" : { "ec2:Vpc" : "vpcArn" }}
```

- Action: `ec2:CreateNetworkInterfacePermission` on `all AWS resources that match the following Condition:`

```
{ "StringLike" : { "ec2:ResourceTag/outposts:private-connectivity-resourceId" :
 "OutpostID" }} and { "StringEquals" : { "ec2:Vpc" : "vpcArn" }}
```

- Action: `ec2:CreateTags` on `all AWS resources that match the following Condition:`

```
{ "StringLike" : { "aws:RequestTag/outposts:private-connectivity-resourceId" :
 "{{OutpostId}}*"}}
```

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

## Creating a service-linked role for AWS Outposts

You don't need to manually create a service-linked role. When you configure private connectivity for your Outpost in the AWS Management Console, AWS Outposts creates the service-linked role for you.

## Editing a service-linked role for AWS Outposts

AWS Outposts does not allow you to edit the AWSServiceRoleForOutposts_*OutpostID* service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

## Deleting a service-linked role for AWS Outposts

If you no longer require a feature or service that requires a service-linked role, we recommend that you delete that role. That way you avoid having an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

**Note**

If the AWS Outposts service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

**Warning**

You must delete your Outpost before you can delete the AWSServiceRoleForOutposts_*OutpostID* service-linked role. The following procedure deletes your Outpost.

Before you begin, make sure that your Outpost is not being shared using AWS Resource Access Manager (AWS RAM). For more information, see Unsharing a shared Outpost resource.

**To delete AWS Outposts resources used by the AWSServiceRoleForOutposts_*OutpostID***

- Contact AWS Enterprise Support to delete your Outpost.

**To manually delete the service-linked role using IAM**

Use the IAM console, the AWS CLI, or the AWS API to delete the AWSServiceRoleForOutposts_*OutpostID* service-linked role. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

### Supported Regions for AWS Outposts service-linked roles

AWS Outposts supports using service-linked roles in all of the Regions where the service is available. For more information, see AWS Outposts endpoints and quotas.

## AWS managed policies for AWS Outposts

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

### AWS managed policy: AWSOutpostsServiceRolePolicy

This policy is attached to a service-linked role that allows AWS Outposts to perform actions on your behalf. For more information, see Using service-linked roles.

## AWS managed policy: AWSOutpostsPrivateConnectivityPolicy

This policy is attached to a service-linked role that allows AWS Outposts to perform actions on your behalf. For more information, see Using service-linked roles.

## AWS managed policy: AWSOutpostsAuthorizeServerPolicy

Use this policy to grant the permissions required to authorize Outpost server hardware in your on-premises network. For more information, see Grant permission.

This policy includes the following permissions.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "outposts:StartConnection",
        "outposts:GetConnection"
      ],
      "Resource": "*"
    }
  ]
}
```

## AWS Outposts updates to AWS managed policies

View details about updates to AWS managed policies for AWS Outposts since this service began tracking these changes.

| Change | Description | Date |
| --- | --- | --- |
| AWSOutpostsAuthorizeServerPolicy – New policy | AWS Outposts added a policy that grants permissions to authorize Outpost server hardware in your on-premises network. | January 4, 2023 |

| Change | Description | Date |
| --- | --- | --- |
| AWS Outposts started tracking changes | AWS Outposts started tracking changes for its AWS managed policies. | December 03, 2019 |

# Infrastructure security in AWS Outposts

As a managed service, AWS Outposts is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access AWS Outposts through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

For more information about the infrastructure security provided for the EC2 instances and EBS volumes running on your Outpost, see Infrastructure Security in Amazon EC2.

VPC Flow Logs function the same way as they do in an AWS Region. This means that they can be published to CloudWatch Logs, Amazon S3, or to Amazon GuardDuty for analysis. Data needs to be sent back to the Region for publication to these services, so it is not visible from CloudWatch or other services when the Outpost is in a disconnected state.

# Resilience in AWS Outposts

AWS Outposts is designed to be highly available. Outpost racks are designed with redundant power and networking equipment. For additional resilience, we recommend that you provide dual power sources and redundant network connectivity for your Outpost.

For high availability, you can order additional Outposts servers. Outpost capacity configurations are designed to operate in production environments, and support N+1 instances for each instance family when you provision the capacity to do so. AWS recommends that you allocate sufficient additional capacity for your mission-critical applications to enable recovery and failover if there is an underlying host issue. You can use the Amazon CloudWatch capacity availability metrics and set alarms to monitor the health of your applications, create CloudWatch actions to configure automatic recovery options, and monitor the capacity utilization of your Outposts over time.

When you create an Outpost, you select an Availability Zone from an AWS Region. This Availability Zone supports control plane operations such as responding to API calls, monitoring the Outpost, and updating the Outpost. To benefit from the resiliency provided by Availability Zones, you can deploy applications on multiple Outposts, each attached to a different Availability Zone. This enables you to build additional application resilience and avoid a dependence on a single Availability Zone. For more information about Regions and Availability Zones, see AWS Global Infrastructure.

Outposts servers include instance store volumes but do not support Amazon EBS volumes. The data on instance store volumes persists after an instance reboot but does not persist after instance termination. To retain the long-term data on your instance store volumes beyond the lifetime of the instance, be sure to back up the data to persistent storage, such as an Amazon S3 bucket or a network storage device in your on-premises network.

# Compliance validation for AWS Outposts

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- Architecting for HIPAA Security and Compliance on Amazon Web Services – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

  > **Note**
  >
  > Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.

- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

# Monitor your Outpost

AWS Outposts integrates with the following services that offer monitoring and logging capabilities:

**CloudWatch metrics**

Use Amazon CloudWatch to retrieve statistics about data points for your Outposts as an ordered set of time series data, known as *metrics*. You can use these metrics to verify that your system is performing as expected. For more information, see CloudWatch metrics for AWS Outposts.

**CloudTrail logs**

Use AWS CloudTrail to capture detailed information about the calls made to AWS APIs. You can store these calls as log files in Amazon S3. You can use these CloudTrail logs to determine such information as which call was made, the source IP address where the call came from, who made the call, and when the call was made.

The CloudTrail logs contain information about the calls to API actions for AWS Outposts. They also contain information for calls to API actions from services on an Outpost, such as Amazon EC2 and Amazon EBS. For more information, see AWS Outposts information in CloudTrail.

**VPC Flow Logs**

Use VPC Flow Logs to capture detailed information about the traffic going to and from your Outpost and within your Outpost. For more information, see VPC Flow Logs in the *Amazon VPC User Guide*.

**Traffic Mirroring**

Use Traffic Mirroring to copy and forward network traffic from Outpost to out-of-band security and monitoring appliances in Outpost. You can use the mirrored traffic for content inspection, threat monitoring, or troubleshooting. For more information, see *Traffic Mirroring Guide* for Amazon Virtual Private Cloud.

**AWS Health Dashboard**

The AWS Health Dashboard displays information and notifications that are initiated by changes in the health of AWS resources. The information is presented in two ways: on a dashboard that shows recent and upcoming events organized by category, and in a full event log that shows all events from the past 90 days. For example, a connectivity issue on the service link would initiate

an event that would appear on the dashboard and event log, and remain in the event log for 90 days. A part of the AWS Health service, AWS Health Dashboard requires no setup and can be viewed by any user that is authenticated in your account. For more information, see Getting started with the AWS Health Dashboard.

# CloudWatch metrics for AWS Outposts

AWS Outposts publishes data points to Amazon CloudWatch for your Outposts. CloudWatch enables you to retrieve statistics about those data points as an ordered set of time series data, known as *metrics*. Think of a metric as a variable to monitor, and the data points as the values of that variable over time. For example, you can monitor the instance capacity available to your Outpost over a specified time period. Each data point has an associated timestamp and an optional unit of measurement.

You can use metrics to verify that your system is performing as expected. For example, you can create a CloudWatch alarm to monitor the `ConnectedStatus` metric. If the average metric is less than 1, CloudWatch can initiate an action, such as sending a notification to an email address. You can then investigate potential on-premises or uplink networking issues that might be impacting the operations of your Outpost. Common issues include recent on-premises network configuration changes to firewall and NAT rules, or internet connection issues. For `ConnectedStatus` issues, we recommend verifying connectivity to the AWS Region from within your on-premises network, and contacting AWS Support if the problem persists.

For more information about creating a CloudWatch alarm, see Using Amazon CloudWatch Alarms in the *Amazon CloudWatch User Guide*. For more information about CloudWatch, see the Amazon CloudWatch User Guide.

**Contents**



## Outpost metrics

The `AWS/Outposts` namespace includes the following metrics.

`ConnectedStatus`

The status of an Outpost's service link connection. If the average statistic is less than 1, the connection is impaired.

**Unit**: Count

**Maximum resolution**: 1 minute

**Statistics**: The most useful statistic is `Average`.

**Dimensions**: `OutpostId`

`CapacityExceptions`

The number of insufficient capacity errors for instance launches.

**Unit**: Count

**Maximum resolution**: 5 minutes

**Statistics**: The most useful statistics are `Maximum` and `Minimum`.

**Dimensions**: `InstanceType` and `OutpostId`

`InstanceFamilyCapacityAvailability`

The percentage of instance capacity available. This metric does not include capacity for any Dedicated Hosts configured on the Outpost.

**Unit**: Percent

**Maximum resolution**: 5 minutes

**Statistics**: The most useful statistics are `Average` and `pNN.NN` (percentiles).

**Dimensions**: `InstanceFamily` and `OutpostId`

`InstanceFamilyCapacityUtilization`

The percentage of instance capacity in use. This metric does not include capacity for any Dedicated Hosts configured on the Outpost.

**Unit**: Percent

**Maximum resolution**: 5 minutes

**Statistics**: The most useful statistics are `Average` and `pNN.NN` (percentiles).

**Dimensions**: `Account`, `InstanceFamily`, and `OutpostId`

`InstanceTypeCapacityAvailability`

The percentage of instance capacity available. This metric does not include capacity for any Dedicated Hosts configured on the Outpost.

**Unit**: Percent

**Maximum resolution**: 5 minutes

**Statistics**: The most useful statistics are `Average` and `pNN.NN` (percentiles).

**Dimensions**: `InstanceType` and `OutpostId`

`InstanceTypeCapacityUtilization`

The percentage of instance capacity in use. This metric does not include capacity for any Dedicated Hosts configured on the Outpost.

**Unit**: Percent

**Maximum resolution**: 5 minutes

**Statistics**: The most useful statistics are `Average` and `pNN.NN` (percentiles).

**Dimensions**: `Account`, `InstanceType`, and `OutpostId`

`UsedInstanceType_Count`

The number of instance types that are currently in use, including any instance types used by managed services such as Amazon Relational Database Service (Amazon RDS) or Application Load Balancer. This metric does not include capacity for any Dedicated Hosts configured on the Outpost.

**Unit**: Count

**Maximum resolution**: 5 minutes

**Dimensions**: `Account`, `InstanceType`, and `OutpostId`

`AvailableInstanceType_Count`

The number of available instance types. This metric does not include capacity for any Dedicated Hosts configured on the Outpost.

**Unit**: Count

**Maximum resolution**: 5 minutes

**Dimensions**: `InstanceType` and `OutpostId`

`AvailableReservedInstances`

The number of instances available on the Outpost for On-Demand Capacity Reservations (ODCR). This metric does not measure Amazon EC2 Reserved Instances.

**Unit**: Count

**Maximum resolution**: 5 minutes

**Dimensions**: `InstanceType` and `OutpostId`

`UsedReservedInstances`

The number of instances available on the Outpost for On-Demand Capacity Reservations (ODCR). This metric does not measure Amazon EC2 Reserved Instances.

**Unit**: Count

**Maximum resolution**: 5 minutes

**Dimensions**: `InstanceType` and `OutpostId`

`TotalReservedInstances`

The number of instances available on the Outpost for On-Demand Capacity Reservations (ODCR). This metric does not measure Amazon EC2 Reserved Instances.

**Unit**: Count

**Maximum resolution**: 5 minutes

**Dimensions**: `InstanceType` and `OutpostId`

## Outpost metric dimensions

To filter the metrics for your Outpost, use the following dimensions.

| Dimension | Description |
| --- | --- |
| `Account` | The account or service using the capacity. |
| `InstanceFamily` | The instance family. |
| `InstanceType` | The instance type. |
| `OutpostId` | The ID of the Outpost. |
| `VolumeType` | The EBS volume type. |
| `VirtualInterfaceId` | The ID of the local gateway or service link Virtual Interface (VIF). |
| `VirtualInterfaceGroupId` | The ID of the virtual interface group for the local gateway Virtual Interface (VIF). |

## View CloudWatch metrics for your outpost

You can view the CloudWatch metrics for your load balancers using the CloudWatch console.

### To view metrics using the CloudWatch console

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Select the **Outposts** namespace.
4. (Optional) To view a metric across all dimensions, enter its name in the search box.

### To view metrics using the AWS CLI

Use the following list-metrics command to list the available metrics.

```
aws cloudwatch list-metrics --namespace AWS/Outposts
```

**To get the statistics for a metric using the AWS CLI**

Use the following get-metric-statistics command to get statistics for the specified metric and dimension. CloudWatch treats each unique combination of dimensions as a separate metric. You can't retrieve statistics using combinations of dimensions that were not specially published. You must specify the same dimensions that were used when the metrics were created.

```
aws cloudwatch get-metric-statistics --namespace AWS/Outposts \
--metric-name InstanceTypeCapacityUtilization --statistics Average  --period 3600 \
--dimensions Name=OutpostId,Value=op-01234567890abcdef \
Name=InstanceType,Value=c5.xlarge \
--start-time 2019-12-01T00:00:00Z --end-time 2019-12-08T00:00:00Z
```

# Log AWS Outposts API calls using AWS CloudTrail

AWS Outposts is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Outposts. CloudTrail captures all API calls for AWS Outposts as events. The calls captured include calls from the AWS Outposts console and code calls to the AWS Outposts API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an S3 bucket, including events for AWS Outposts. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to AWS Outposts, the IP address from which the request was made, who made the request, when it was made, and additional details.

For more information about CloudTrail, see the AWS CloudTrail User Guide.

## AWS Outposts information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS Outposts, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail event history.

For an ongoing record of events in your AWS account, including events for AWS Outposts, create a trail. A *trail* enables CloudTrail to deliver log files to an S3 bucket in the parent AWS Region. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the S3 bucket that you

specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for creating a trail
- CloudTrail Supported services and integrations
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple Regions and Receiving CloudTrail log files from multiple accounts

All AWS Outposts actions are logged by CloudTrail. They are documented in the *AWS Outposts API Reference*. For example, calls to the `CreateOutpost`, `GetOutpostInstanceTypes`, and `ListSites` actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine whether the request was made:

- With root or user credentials.
- With temporary security credentials for a role or federated user.
- By another AWS service.

For more information, see the CloudTrail userIdentity element.

## Understanding AWS Outposts log file entries

A trail is a configuration that enables delivery of events as log files to an S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source. It includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `CreateOutpost` action.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE:jdoe",
```

```
        "arn": "arn:aws:sts::111122223333:assumed-role/example/jdoe",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/example",
                "accountId": "111122223333",
                "userName": "example"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-08-14T16:28:16Z"
            }
        }
    },
    "eventTime": "2020-08-14T16:32:23Z",
    "eventSource": "outposts.amazonaws.com",
    "eventName": "SetSiteAddress",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "XXX.XXX.XXX.XXX",
    "userAgent": "userAgent",
    "requestParameters": {
        "SiteId": "os-123ab4c56789de01f",
        "Address": "***"
    },
    "responseElements": {
        "Address": "***",
        "SiteId": "os-123ab4c56789de01f"
    },
    "requestID": "1abcd23e-f4gh-567j-klm8-9np01q234r56",
    "eventID": "1234a56b-c78d-9e0f-g1h2-34jk56m7n890",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

# Outpost maintenance

Under the shared responsibility model, AWS is responsible for the hardware and software that run AWS services. This applies to AWS Outposts, just as it does to an AWS Region. For example, AWS manages security patches, updates firmware, and maintains the Outpost equipment. AWS also monitors the performance, health, and metrics for your Outpost and determines whether any maintenance is required.

> **Warning**
>
> Data on instance store volumes is lost if the underlying disk drive fails, or if the instance terminates. To prevent data loss, we recommend that you back up your long-term data on instance store volumes to persistent storage, such as an Amazon S3 bucket or a network storage device in your on-premises network.

**Contents**



## Hardware maintenance

If AWS detects an irreparable issue with hardware hosting Amazon EC2 instances running on your Outpost, we will notify the owner of the Outpost and the owner of the instances that the affected instances are scheduled for retirement. For more information, see Instance retirement in the *Amazon EC2 User Guide*.

AWS terminates the affected instances on the instance retirement date. The data on instance store volumes does not persist after instance termination. Therefore, it is important that you take action before the instance retirement date. First, transfer your long-term data from the instance store volumes for each affected instance to persistent storage, such as an Amazon S3 bucket or a network storage device in your network.

A replacement server will be shipped to the Outpost site. Then, do the following:

- Remove the network and power cables from the irreparable server and if necessary remove it from your rack.
- Install the replacement server in the same location. Follow the installation instructions in Outpost server installation.
- Pack the irreparable server to AWS in the same packaging that the replacement server arrived in.
- Use the pre-paid return shipment label that is available in the console attached to the order configuration details or the replacement server order.
- Return the server to AWS. For more information, see Return an AWS Outposts server.

# Firmware updates

Updating the Outpost firmware does not typically affect the instances on your Outpost. In the rare case that we need to reboot the Outpost equipment to install an update, you will receive an instance retirement notice for any instances running on that capacity.

# Best practices for AWS Outposts power and network events

As stated in the AWS Service Terms for AWS Outposts customers, the facility where the Outposts equipment is located must meet the minimum power and network requirements to support the installation, maintenance, and use of the Outposts equipment. An Outposts rack can operate correctly only when power and network connectivity is uninterrupted.

## Power events

With complete power outages, there is an inherent risk that an AWS Outposts resource may not return to service automatically. In addition to deploying redundant power and backup power solutions, we recommend that you do the following in advance to mitigate the impact of some of the worst-case scenarios:

- Move your services and applications off the Outposts equipment in a controlled fashion, using DNS-based or off-rack load-balancing changes.
- Stop containers, instances, databases in an ordered incremental fashion and use the reverse order when restoring them.
- Test plans for the controlled moving or stopping of services.
- Back-up critical data and configurations and store it outside the Outposts.
- Keep power downtimes to a minimum.

- Avoid repeated switching of the power feeds (off-on-off-on) during the maintenance.
- Allow for extra time within the maintenance window to deal with the unexpected.
- Manage the expectations of your users and customers by communicating a wider maintenance window time-frame than you would normally need.

## Network connectivity events

The service link connection between your Outpost and the AWS Region or Outposts home Region will typically automatically recover from network interruptions or issues that may occur in your upstream corporate network devices or in the network of any third party connectivity provider once the network maintenance is completed. During the time the service link connection is down, your Outposts operations are limited to local network activities. For more information, see the question *What happens when my facility's network connection goes down?* on the AWS Outposts rack FAQs page.

If the service link is down because of an on-site power issue or the loss of network connectivity, the AWS Health Dashboard sends a notification to the account that owns the Outposts. Neither you nor AWS can suppress the notification of a service link interruption, even if the interruption is expected. For more information, see Getting started with your AWS Health Dashboard in the *AWS Health User Guide*.

In the case of a planned service maintenance that will affect network connectivity, take the following proactive steps to limit the impact of potential problematic scenarios:

- If your Outposts rack connects to the parent AWS Region through Internet or public Direct Connect, then in advance of a planned maintenance, capture a trace-route. Having a working (pre-network-maintenance) network path and a problematic (post-network-maintenance) network path to identify the differences would help in troubleshooting. If you escalate a post-maintenance issue to AWS or your ISP, you can include this information.

  Capture a trace-route between:

  - The public IP addresses at the Outposts location and the IP address returned by the `outposts.region.amazonaws.com`. Replace *`region`* with the name of the parent AWS Region.
  - Any instance in the parent Region with public Internet connectivity and the public IP addresses at the Outposts location.

- If you are in control of the network maintenance, limit the duration of downtime for the service link. Include a step in your maintenance process that verifies that the network has recovered.
- If you are not in control of the network maintenance, monitor the service link downtime with respect to the announced maintenance window and escalate early to the party in charge of the planned network maintenance if the service link is not back up at the end of the announced maintenance window.

### Resources

Here are some monitoring related resources that can provide reassurance that the Outposts is operating normally after a planned or unplanned power or network event:

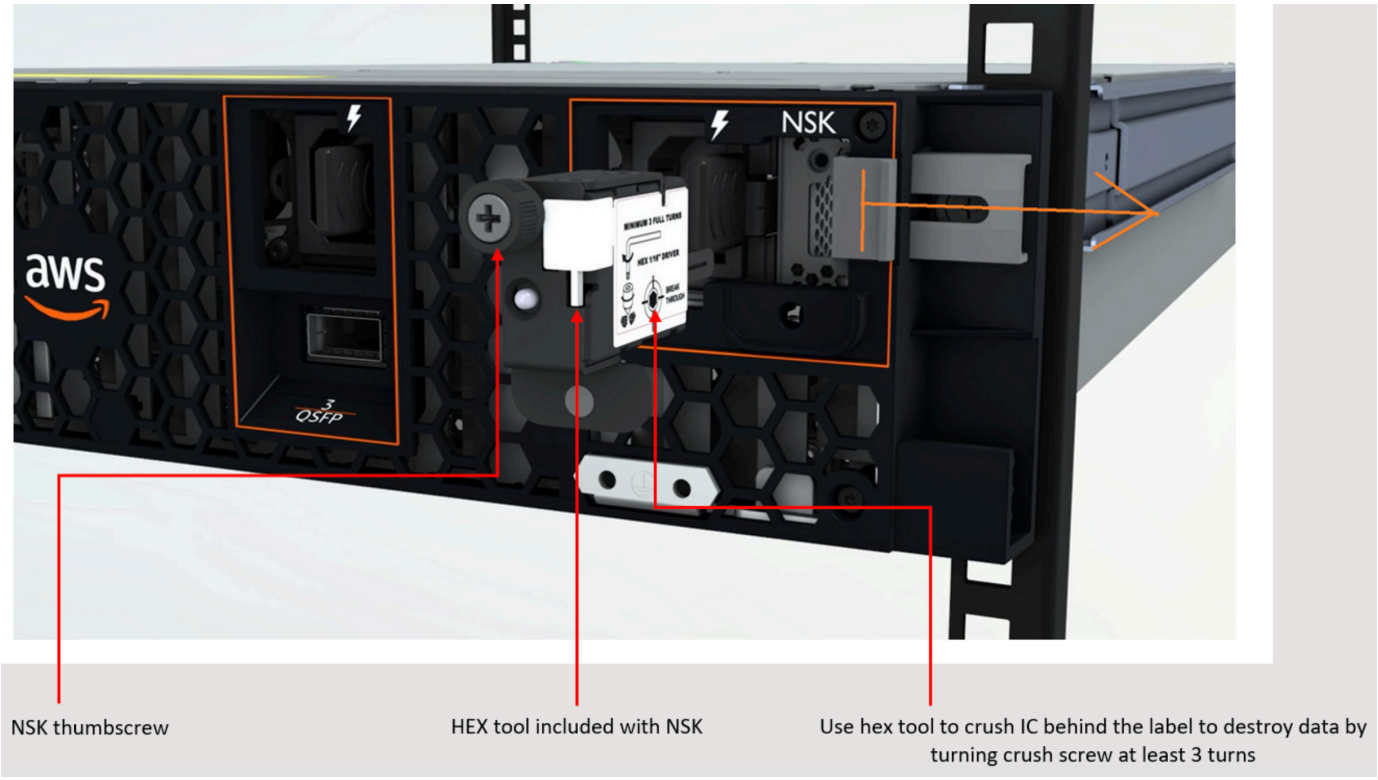
- The AWS blog Monitoring best practices for AWS Outposts covers observability and event management best practices specific to Outposts.
- The AWS blog Debugging tool for network connectivity from Amazon VPC explains the *AWSSupport-SetupIPMonitoringFromVPC* tool. This tool is an AWS Systems Manager document (SSM document) that creates an Amazon EC2 Monitor Instance in a subnet specified by you and monitors target IP addresses. The document runs ping, MTR, TCP trace-route and trace-path diagnostic tests and stores the results in Amazon CloudWatch Logs which can be visualized in a CloudWatch dashboard (e.g. latency, packet loss). For Outposts monitoring, the Monitor Instance should be in one subnet of the parent AWS Region and configured to monitor one or more of your Outpost instances using its private IP(s) - this will provide packet loss graphs and latency between AWS Outposts and the parent AWS Region.
- The AWS blog Deploying an automated Amazon CloudWatch dashboard for AWS Outposts using AWS CDK describes the steps involved in deploying an automated dashboard.
- If you have questions or need more information, see Creating a support case in the *AWS Support User Guide*.

## Cryptographically shred server data

The Nitro Security Key (NSK) is required to decrypt data on the server. When you return the server to AWS, either because you are replacing the server or discontinuing the service, you can destroy the NSK to cryptographically shred the data on the server.

**To cryptographically shred data on the server**

1. Remove the NSK from the server before shipping the server back to AWS.
2. Ensure that you have the correct NSK that shipped with the server.
3. Remove the small hex tool / Allen wrench from under the sticker.
4. Use the hex tool to turn the small screw under the sticker three full turns. This action destroys the NSK and cryptographically shreds all data on the server.



# AWS Outposts end-of-term options

At the end of your AWS Outposts term, you have three options:

- Renew your subscription and keep your existing Outpost.
- End your subscription and return your Outpost server.
- Convert to a month-to-month subscription and keep your existing Outpost server.

If you do not indicate that you want to renew your subscription or return your Outpost server, you will be converted to a month-to-month subscription.

**Topics**



# Renew your subscription

**To renew your subscription and keep your existing Outpost server:**

Complete the following steps at least **30 days** before your Outpost's term ends:

1. Sign in to the AWS Support Center Console.
2. Choose **Create case**.
3. Choose **Account and billing**.
4. For **Service**, choose **Billing**.
5. For **Category**, choose **Other Billing Questions**.
6. For **Severity**, choose **Important question**.
7. Choose **Next step: Additional information**.
8. On the **Additional information** page, for **Subject**, enter your request to renew such as **`Renew my Outpost subscription`**.
9. For **Description**, enter one of the following payment options:
    - No upfront

- Partial upfront
- All upfront

For pricing, see AWS Outposts servers pricing. You can also request a price quote.

10. Choose **Next step: Solve now or contact us**.
11. On the **Contact us** page, choose your preferred language.
12. Choose your preferred contact method.
13. Review your case details and then choose **Submit**. Your case ID number and summary appear.

AWS Customer Support will initiate the subscription renewal process. Your new subscription will start the day after your current subscription ends.

## End your subscription and return the server

> **Important**
>
> AWS cannot begin the return process until you have completed the following procedure. We cannot stop the return process after you have opened a support case to end your subscription.

**To end your subscription:**

Complete the following steps at least **30 days** before your Outpost's term ends:

1. Sign in to the AWS Support Center Console.
2. Choose **Create case**.
3. Choose **Account and billing**.
4. For **Service**, choose **Billing**.
5. For **Category**, choose **Other Billing Questions**.
6. For **Severity**, choose **Important question**.
7. Choose **Next step: Additional information**.
8. On the **Additional information** page, for **Subject**, enter a clear request, such as **`End my Outpost subscription`**.

9. For **Description**, enter the date you wish to end your subscription.
10. Choose **Next step: Solve now or contact us**.
11. On the **Contact us** page, choose your preferred language.
12. Choose your preferred contact method.
13. If necessary, back up any instances and instance data present on your server.
14. Terminate instances launched on your server.
15. Review your case details and then choose **Submit**. Your case ID number and summary appear.
16. Do NOT power down or disconnect the server from the network until instructed to do so in the support case.

To return your AWS Outposts server, follow the procedures on Return an AWS Outposts server.

## Convert to a month-to-month subscription

To convert to a month-to-month subscription and keep your existing Outpost server, no action is needed. If you have questions, open a billing support case.

Your Outpost will be renewed on a monthly basis at the rate of the **No Upfront** payment option that corresponds to your AWS Outposts configuration. Your new monthly subscription will start the day after your current subscription ends.

# Quotas for AWS Outposts

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, but not for all quotas.

To view the quotas for AWS Outposts, open the Service Quotas console. In the navigation pane, choose **AWS services**, and select **AWS Outposts**.

To request a quota increase, see Requesting a quota increase in the *Service Quotas User Guide*.

Your AWS account has the following quotas related to AWS Outposts.

| Resource | Default | Adjustable | Comments |
|---|---|---|---|
| Outpost sites | 100 | Yes | An Outpost site is the customer managed physical building where you power and attach your Outpost equipment to the network.<br><br>You can have 100 Outposts sites in each Region of your AWS account. |
| Outposts per site | 10 | Yes | AWS Outposts includes hardware and virtual resources, known as Outposts. This quota limits your Outpost virtual resources.<br><br>You can have 10 Outposts in each Outpost site. |

## AWS Outposts and the quotas for other services

AWS Outposts relies on the resources of other services and those services may have their own default quotas. For example, your quota for local network interfaces comes from the Amazon VPC quota for network interfaces.

# Document history

The following table describes important changes to the *AWS Outposts User Guide*.

| Change | Description | Date |
| --- | --- | --- |
| Capacity management | You can modify the default capacity configuration for your new Outposts order. | April 16, 2024 |
| End-of-term options for AWS Outposts servers | At the end of your AWS Outposts term, you can renew, end, or convert your subscription. | August 1, 2023 |
| Created AWS Outposts User Guide for Outposts servers | AWS Outposts User Guide broke into separate guides for rack and servers. | September 14, 2022 |
| Placement groups on AWS Outposts | Placement groups that use a spread strategy can distribute instances across hosts. | June 30, 2022 |
| Dedicated Hosts on AWS Outposts | You can now use Dedicated Hosts on Outposts. | May 31, 2022 |
| Introducing Outpost servers | Added Outposts servers, a new AWS Outposts form factor. | November 30, 2021 |